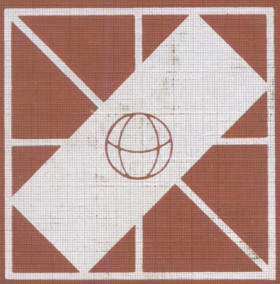
computer message systems

edited by
r.p. uhlig



IFIP

north-holland

computer message systems

uhlig
editor

COMPUTER MESSAGE SYSTEMS

IFIP Symposium on
Computer Message Systems
Ottawa, Canada, 6-8 April, 1981

sponsored by
International Federation for Information Processing (IFIP)

organized by
IFIP TC-6, Working Group 6.5
International Computer Message Services

program committee
Ronald Uhlig, Chairman (Canada), Andre Danthine (Belgium), David Twyver (Canada), Najah Naffah (France), Klaus Wimmer (Germany), Tibor Szentivanyi (Hungary), Giuseppe Attardi (Italy), Peter Schinker (Switzerland), Donald Davies (UK), David Farber (USA), Theordore Myer (USA)

organizing committee
Peter Trau, Chairman, Bell-Northern Research, Canada
Bill Newman, Treasurer, Bell-Northern Research, Canada
Constance Clapp, Secretary, Bell-Northern Research, Canada

NORTH-HOLLAND PUBLISHING COMPANY
AMSTERDAM • NEW YORK • OXFORD

COMPUTER MESSAGE SYSTEMS

Proceedings of the IFIP TC-6 International Symposium on
Computer Message Systems
Ottawa, Canada, 6-8 April, 1981

edited by

RONALD P. UHLIG
Bell-Northern Research Limited
Ottawa, Canada

NORTH-HOLLAND PUBLISHING COMPANY
AMSTERDAM • NEW YORK • OXFORD

© IFIP, 1981

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owner.

ISBN: 0444852536

Published by:

NORTH-HOLLAND PUBLISHING COMPANY – AMSTERDAM • NEW YORK • OXFORD

Sole distributors for the U.S.A. and Canada:

ELSEVIER NORTH-HOLLAND, INC.
52 Vanderbilt Avenue
New York, N.Y. 10017

Library of Congress Cataloging in Publication Data

IFIP TC-6 International Symposium on Computer Message Systems (1981 : Ottawa, Ont.)
Computer message systems.

1. Data transmission systems--Congresses. 2. Computer networks--Congresses. 3. Information networks--Congresses. I. Uhlig, Ronald P., 1940- II. International Federation for Information Processing. Technical Committee 6.
TK5105.I33 1981 001.64'4 81-11010
ISBN 0-444-85253-6 (U.S.) AACR2

PRINTED IN THE NETHERLANDS

PREFACE

This symposium was a historic meeting. It was the first time that a major meeting was devoted exclusively to the topic of computer message systems. The symposium was one in a series of annual in-depth symposia on various areas of data communications sponsored by I.F.I.P. Technical Committee 6, Data Communications. This 1981 in-depth symposium was organized on behalf of TC-6 by Working Group 6.5, International Computer Message Services.

I.F.I.P., the International Federation for Information Processing, is made up of the national computing societies of many nations. I.F.I.P. carries out its business through acitivities organized by its 9 technical committees and their working groups, as well as through sponsoring a triennial World Computer Congress.

On the recommendation of TC-6, Working Group (WG) 6.5 was formally approved by the I.F.I.P. General Assembly in September 1979. The purpose of the Working Group is to bring together people from around the world who are interested in working together on the many challenges of computer based human to human messaging systems. WG 6.5 organized two international workshops on computer messaging in Montreal, Canada and Bonn, Federal Republic of Germany prior to this symposium. In addition, approximately 20 other meetings of smaller groups of people have been held under the auspices of WG 6.5 since it was formed.

Over 200 participants from 20 countries took part in three days of presentations and discussions at this 1981 symposium. The topics discussed cut across the entire spectrum of technical, human factors, and policy and regulatory issues associated with computer messaging. Other conferences have organized one or several sessions on computer messaging, but never before has an entire conference been devoted to computer messaging.

The high quality of the papers contained in this volume not only reflect well on the individual authors, but are also indicative of the timeliness of this symposium. The large number of papers submitted, and the many inquiries received, show that computer messaging has become a major topic of world-wide interest. Regretably, it was impossible to accept all papers submitted, but I.F.I.P. TC-6 would like to express thanks to all who went to the effort of submitting papers. As Chairman of the Program Committee, and on behalf of TC-6, I want to express particular thanks and gratitude to the authors who wrote the papers contained in this volume. 33 papers from 12 countries were arranged into 15 sessions for this symposium.

A great deal of work goes into organizing a symposium of this magnitude. It is impossible to thank all who contributed, but I would like to thank all the members of the Program Committee and the Organizing Committee. Members of the Program Committee were critical to success in getting broad international participation, and I would particularly like to thank Peter Trau, Chairman of Organizing Committee, Constance Clapp, Secretary, and William Newman, Treasurer, who gave unsparingly of their time. On behalf of the I.F.I.P. TC-6, I would also like to thank Bell-Northern Research and Bell Canada without whose generous support it would have been impossible to hold this symposium.

Ronald P. Uhlig
Chairman
Program Committee

TABLE OF CONTENTS

KEYNOTE SESSION : Chairman - R. Uhlig

What's the Message
G.B. THOMPSON 3

MESSAGE SYSTEMS FOR DEVELOPING NATIONS :
Chairman - R. Uhlig

Computer Message Systems for Developing Nations -
A Design Exercise
P.P. GUPTA and S. RAMANI 9

VOICE MESSAGING : Chairman - D. Farber

A Voice Message System
D. COHEN 17

An Experimental Voice-Message System
Controlled by Word-Recognition
P. MERMELSTEIN and V. GUPTA 29

FACSIMILE MESSAGING AND EXTENSIONS :
Chairman - K. Ono

Network Architecture of Advanced Facsimile
Intelligent Communication System
K. WATANABE, Y. KIRIHARA, T. ENDO
and T. KAMAE 37

The Integrated Services Digital Network (ISDN) and its
Use for Text and Data Communication
P. BOCKER and P.R. GERKE 53

USER REQUIREMENTS FOR COMPUTER MESSAGING :
Chairman - E. Feinler

User Experience and Evolving Design in a Local Electronic Mail System
J. BRUDER, M. MOY, A. MUELLER and R. DANIELSON 69

Computer Mediated Messaging for Interactive Purposes
W. DZIDA 79

COMPUTER CONFERENCING : Chairman - A. Danthine

The Effect of Structure, Task, and Individual Attributes on Concensus in Computerized Conferences
S.R. HILTZ, M. TUROFF and K. JOHNSON 91

Computerized Message Sending and Teleconferencing in an International Environment - Present and Future
I. SEBESTYEN 103

The VMSHARE Computer Conferencing Facility
C. DANEY 115

Creating an Adaptive Computerized Conferencing System on Unix
M.M.L. PEARSON and J.E. KULP 129

COMPUTER MESSAGE SYSTEM DESIGN : Chairman - D. Twyver

The Message Processing System in the DS-1 International Packet Switching System
T. NAMIKI, K. SODEYAMA and T. MARUYAMA 147

Service Definitions in a Computer Based Mail Environment
P. SCHICKER 159

Active Message Processing: Messages as Messengers
J. VITTAL 175

COMPUTER MESSAGE SYSTEM PROTOCOLS :
Chairman - N. Naffah

Design of a Message Format Standard
D. DEUTSCH 199

A Methodology for the Specification of a Message Transport System
G. v.BOCHMANN and J.R. PICKENS 221

On Reliable Transport Connections for Exchanging Mail Items
F.H. VOGT 233

TELETEX : Chairman - I. Cunningham

Protocol Architecture and its Application for CCITT Standardized Services
W. STAUDINGER 247

Teletex and it's Protocols
G.A. ROUTHORN and P.A. CARRUTHERS 259

The Teletex Service in the Federal Republic of Germany
R. RÜGGEBERG 285

NAMING/ADDRESSING ISSUES : Chairman - P. Schicker

Addressing and Directory Systems for Large Computer Mail Systems
J.J. GARCIA-LUNA and F.F. KUO 297

Interconnection of Electronic Mail Systems - A Proposal on Naming, Addressing and Routing
I.H. KERR 315

COMPUTER MESSAGE SYSTEM IMPLEMENTATION :
Chairman - P. Gupta

MSG - A Simple Message System
J. VITTAL 329

An Experimental Message Computer System
between Universities in Brazil
L.M.R. TAROUCO 345

MAIL2K: A Computer Mail Package for
Local Applications
E. MOTOLESE, R. FONDI, E. SALVADORI and
M. VIGNALI 353

CORPORATE MESSAGE SYSTEMS : Chairman - T. Myer

An International Text Message Service for
Corporate Use
I. ZANDLEVEN 363

The Design and Service Impact of Cocos, An Electronic
Office System
N.W. DAWES, S.J. HARRIS, M.I. MAGOON,
S.J. MAVEETY and D.J. PETTY 373

MESSAGE SYSTEMS AND OFFICE AUTOMATION :
Chairman - H. Schuchmann

The Role of Computer Mail in Office Automation
E. STEFFERUD and J. MCHUGH 387

Research on the Impact of Office Information
Communication Systems
D. TAPSCOTT 395

SECURITY AND MESSAGE SYSTEMS : Chairman - C. Heitmeyer

Encryption in Computer Networks and Message Systems
W.L. PRICE 413

HUMAN FACTORS IN COMPUTER MESSAGE SYSTEMS :
Chairman - G. Attardi

The User Interface Design Process
H. WILLIAMSON and S. ROHLFS 427

Acceptance Research Strategies in Computer Message Systems
R. HELMREICH 447

Human Factors Requirements for Managerial Use of Computer Message Systems
J. NEWMAN 453

KEYNOTE SESSION

Chairman - R. Uhlig

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP,1981

WHAT'S THE MESSAGE?

by

Gordon B. Thompson,

Bell Northern Research,

Ottawa, Canada

April 6, 1981

Marshall McLuhan told us that the medium is the message. Without striving too hard, we can find many examples from our television experience where the medium if not completely changed, at least significantly coloured the perception of the events that actually occurred.

However, it is not the broadcast media I am interested in exploring today. Let us look at the common media we use for messaging and see how the message can be distorted.

Consider the loudspeaking telephone. This device destroys the telephone's shared acoustic space because of the gain swapping strategy that these products use to control acoustic howl. This is particularly noticable when the loudspeaking telephone is connected with an ordinary telephone. The person using the ordinary set gets the impression he's not being listened to. Whenever he talks, the sounds from the other end get cut off, and he thinks he's talking into nothingness.

The loudspeaking telephone is a strange product. It creates a surround of ordinary telephone users who have a strong dislike for the loudspeaking telephone. For everyone that is sold, it creates 20 people who will never buy one!

People who are in the business of serving the public can be sold this set because it is hands-free. However, using the loudspeaking telephone can significantly influence the impression others have of the person using the unit. This influence can be strongly negative. In spite of this, nobody has sued the telephone company, to my knowledge, for this alteration of character.

The end effect is that one should avoid using the loudspeaking telephone when asking for a favour. Alternatively, if the task is one of disciplining, then try to get the recipient on an ordinary

telephone and use a loudspeaking set at your end. Most effective. After a few exposures, you will just have to announce the call, and the recipient will know the message! The medium has become the message.

The long distance echo suppressor also shatters the telephone's shared acoustic space. The story is told of the litle old lady in Toronto who called her sister in Vancouver to share a cry with her. Our echo suppressors made this an impossibility, and forced them to cry at each other sequentially. They complained. Most users don't because they don't know any better.

The real culprit in terms of doing damage, is the satellite with its 250 millisecond delay time. Only recently have we become aware of how this delay can deleteriously impact the perception conversants have of each other's emotional state. Couples that have spent a lot of time using terrestial facilites can have serious difficulties in their relationship, if one should go overseas, entirely due to the effect of the transmission delay.

The satellite delay can, and has, caused couples to break up. The medium has the power to change a message that was intended to move in one direction to go in a totally opposite one. The medium changes the message.

The effect is so subtle, users don't think that it is the satellite that is doing them in. Curiously, nobody has sued the phone company for alienation of affection. Yet.

What has all this to do with messaging? Plenty. What unexpected alterations will the new messaging systems make to the messages we send on them?

Consider voice messaging systems. Should the voice that is delivered be "toll quality", or should it be a synthesized one? If we look at written communications as analagous, then we ought to suspect that the synthesized voice would win out on the grounds that the typewriter continues to exist in spite of our many Xerox machines that make hand written messages quite easy to send.

Earlier we noted that people rarely complain about the way the telephone distorts their messages. Ignorance was suggested as a reason. If we were to go out and ask 100 people whether or not they would prefer a synthetic computer voice to the sender's voice in a message system, I'm sure 99% would go for the "real thing".

Probably, if you asked people about the future of the ersatz writing that the printing press produced, when it was new, they would respond that it was hard to read and it would never replace the "real thing". With all our strange dialects, I wouldn't be surprised if

computer speech would be preferred in a decade or two. Furthermore, our own voice may reveal more of our inner selves than we might wish to expose. Some people argue that this is why we use the typewriter so much. We give its output authority, and this may also happen with synthesized voice.

I find it curious that, in a world where HIFI is a big business, we don't want HIFI telephones. The common telephone produces something like twenty percent harmonic distortion while our HIFIs produce so little distortion that mere measurement is a problem. McLuhan suggests that this is what keeps the telephone "cool", and that a HIFI telephone would be too "hot", and would convey too much of the affective parts of our speech and would destroy the telephone as we now know it.

Perhaps, if we went to HIFI telephones, we would see the emergence of the "professional telephone talker", analagous to radio and TV's announcer. Such a move would significantly reduce the conviviality of the telephone, in the Ivan Illich sense. Again, the medium has a message. Perhaps we are doing research for voice messaging systems that is literally non-productive, if a widely used and trusted medium is the objective. Just what the optimum characteristics might be to maximize conviviality is unknown. There is clearly no indication that HIFI is a necessary characteristic, and it may be quite the opposite.

In the area of typed messaging, Computer Conferencing, as opposed to simple mail systems, is probably the best understood messaging system because of all the good work that has been done towards developing an insight into how such system interact with people. I fear that much of this work is being ignored while simple messaging systems get installed where one of the more complex conferencing system might better have been adopted. As noted earlier, we can not expect people to readily grasp the subtle differences in view of the almost universal trivial level of knowledge and experience in the nitty gritty of communications. Perhaps this is why a lot of disappointing systems get built.

Electronic Mail Systems give me some major concern. The use of the word "mail" brings with it a lot of baggage, and most certainly people are going to get some surprises because of this. A conventional letter always presents itself to the reader in the same format as it had when it left the writer. In the electronic situation, unless rigid controls are exercised over the terminals allowed on the system, there is no guarantee that the recipient will see the same lay out at all. Designers tell us that the way text is presented can significantly alter the attitude the reader has towards printed text. In electronic mail this variable is left wide open!

Whether a message is received on a CRT or a printing terminal probably influences the thoughtfullness of the recipient's reply. An ephemeral message will likely get a less considered response than a more permanent message, particularly if the message is of any length. Recall Lincoln's comment about a longish letter that he wrote, saying he didn't have time to make it shorter. Now, computer messaging is supposed to speed things up, and if it puts a cost on long messages in terms of recipient thoughtfullness, then it may be more of a time waster than a time saver, if we have to be very succinct in our outgoing messages to get the attention we desire.

The fact that our older media are still producing some surprises in terms of how they influence the messages they carry should give us concern that perhaps we don't know enough yet to prescribe the utopian messaging system.

In this brief outline of the way messages can be influenced by the medium, we did not look at how cultural differences can augment and transform this whole area. The Japanese do not have a word that corresponds to our word "communication". I sometimes wonder if we understand our own word. Communications is so important that I suspect it's not a case of using communications as an adjunct to work, but rather working in order to be in a rich communications environment.

MESSAGE SYSTEMS FOR DEVELOPING NATIONS

Chairman - R. Uhlig

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP,1981

COMPUTER MESSAGE SYSTEMS FOR DEVELOPING NATIONS
A DESIGN EXERCISE

P.P. Gupta
Computer Maintenance Corporation
World Trade Centre, Bombay 400 005

S. Ramani
National Centre for Software Development
and Computing Techniques
TIFR, Bombay 400 005

This paper considers computer message systems which can serve large sections of the populations of developing countries. It is possible to define cost effective message systems for this purpose that can, to some extent, reduce the demand for rapid growth of voice communication facilities. Once a computer message system is provided for, a number of technically advanced user-oriented facilities can be added at a relatively low incremental cost. Such facilities, for instance, authentication of origin, machine generated acknowledgements and encryption, are essential for systems being planned in the 80's for use in business and government. Message systems of this type can use existing transmission facilities as well as new low-cost facilities because of the low bit rates and poor signal-to-noise ratios at which they can operate successfully. These systems can play a valuable role in extending communication facilities to sparsely populated rural areas. Weaknesses of the proposed type of message systems include the current predominance of the Roman script and the fact that messaging is not a form of communication usable by large sections of the population in the developing world burdened by illiteracy.

THE PROBLEM

Communication in developing countries poses special problems. Where voice communication equipment supporting one telephone costs ten times the GNP per capita, and the availability of telephones is less than one instrument per hundred of population, one needs to think of genuine alternatives; deferred payment facilities will not suffice.

There are many developing countries that face this communication situation. But it is clear that the problem is going to grow dramatically. Some of them have a mail system that seems to be as speedy as that of industrially developed countries. But, on deeper analysis one finds that the mail volume is of the order of a few items of mail per person per year. Hopefully, economies will grow in a decade to permit the population to have ten times this meagre volume of mail. What will this growth do to the mail systems in developing countries? Can these systems survive such crushing burdens?

Industrialisation, natural economic growth, needs of modern government, and increasing mobility of the population are some of the factors that increase demands for communication services in developing nations. Because of the large populations involved, the investments required even to meet a fraction of the felt need are quite large.

The cost per installed telephone line varies significantly with user density. Costs in rural areas can be higher by a factor of five or more, when compared to costs in a metropolis. It is also well-known in developing countries that returns on communication investments are lower in rural areas, as compared to major cities; this makes low-cost communication alternatives particularly appealing for use in rural areas.

In the industrially advanced countries, message systems are being investigated as an area for marketing new equipment and services. The emphasis is on creating and meeting new needs, such as office automation, computer mediated conferencing, etc. On the other hand, in the developing world it is worth investigating if the growth of message systems can substitute for growth in more traditional communication systems (such as telephony) to some extent, creating real alternatives in communication system options. This orientation is the basis of all considerations that follow.

TECHNOLOGICAL DEVELOPMENTS

Technological development, per se, does not offer an adequate solution. There has been no dramatic new solution to the problem described above. Some developments divide the market place and make problems worse for a part of the population. For instance, introduction of Telex and Subscriber Trunk Dialling tends to take the relatively affluent user organisations away from telegraphy and thus could make telegraphy revenues stagnate or decline. It is likely that in some developing countries telegraphy could quickly enter the vicious circle of lower volumes resulting in a higher cost, which in turn would cause further reductions in volume.

Developments in transmission, such as the use of communication satellites and developments in switching, such as the use of stored-program controlled exchanges do not solve the problem raised here. A net reduction in cost per unit-bandwidth of transmission does not automatically benefit the low-end user of communication systems. Very often, it benefits only the more affluent user, stimulating his consumption of communication services and increasing the range of services available to him.

AN ALTERNATIVE FUTURE

It appears unavoidable that the developing world should seek alternative futures in communication technology. Greater investments in voice communication, adoption of digital switching and use of satellites in conventional ways, are not enough.

What can a technologically up-to-date text-transmission system offer in the way of low-cost public telecommunication? The basic challange it can face is meeting the needs for an order of magnitude, or more, of increase in text transmission facilities from Telex and telegraphy users, and the provision of a new range of services which are by-products of computer-based text handling.

The considerations that follow are primarily technological; operational and administrative points of view are not covered here.

THE PROS AND CONS OF TEXT TRANSMISSION

In some ways, the alternative investigated here is a step backward. Backward, that is, from a one-dimensional advance along the direction of greater reliance on instantaneous two-way voice communication as the basic telecommunication service. There are clearly situations which call for two-way voice communication; situations in which the best computer message system will not suffice. But the bulk of the needs of modern government and business, and a

large fraction of the needs of individuals, by way of inter-city communication are met quite well by text transmission, by standards acceptable to the developing world.

A major objection to text transmission systems could be that they deny their benefits to the illiterate part of the population. But, if initially they are planned as systems providing for a volume of 10 messages per capita per year, this does not pose an impossible problem. The assistance that literate intermediaries offer illiterate users, and the fact that rapid inter-city communication needs are felt much more by literate users, reduce the magnitude of this problem. Increasing literacy in the developing world is another important factor.

A second problem is the predominance of the Roman script in message systems and text transmission systems in general. Dot-matrix and daisy-wheel techniques for printing hard-copies are easily adaptable to other scripts, and key-boards with IC encoders make it easy to provide for key-boarding most human languages. But energetic steps will be required to convert technical possibilities into practical realities.

On the positive side, text transmission systems are free from the misuse that voice systems and even facsimile systems are subject to. Because messages in the form of text have machine-readable addresses, and because messages are archived, there would be much greater discipline in the use of message systems. In comparison, one finds even in developing countries fairly widespread misuse of telephones provided in the offices. In fact, the flood of private calls generated in a communication-starved society make telephone systems very inefficient for use in business or government, by making it necessary to dial twenty times to put one trunk call through.

FRUGALITY IN THE USE OF COMMUNICATION CAPACITY & OTHER BENEFITS

A basic feature of text-based messaging is that the communication capacities required [1] are two to three orders of magnitude below that required for voice communication (around 40 BPS for text key-boarded at 60 words per minute). The fact that a record (hardcopy or computer archived) is available for careful perusal, and for confirmation, is another basic feature. That these messages can be "filed" automatically according to a subject indication field in the message and indexed by date and origin are other valuable features.

THE SIGNIFICANCE OF QUEUEING

That a computer message system can queue up messages is obvious; but, the implications, for instance, on the scale of switching equipment required, are not widely recognised.

As the numbers of users in a system goes up, switching complexity goes up much faster than in linear proportion. When a pair of users wish to communicate with each other, it is obviously desirable that the number of intervening (telephone) exchanges be a minimum. This basic constraint justifies large scale switching equipment, making it desirable to have SPC systems for 10,000 lines or more. Can one handle voice communication requirements of 500,000 subscribers using 300-line or even 1000-line exchanges?

On the other hand, very large message systems can be easily configured in terms of small units, each one handling fifty or a hundred terminals. The connectivity problem is much easier to solve in this case, because there is no need to create links between an arbitrary pair of users in real time. The possibility of using smaller switching equipment opens up a variety of options: faster R&D, possibility of small scale R&D and manufacture, economy of scale

arising from the use of identical switching equipment for urban and rural areas, etc.

The fact that messages are routinely queued has another major implication: utilisation of switching and transmission equipment can be much higher than is possible with voice communication equipment. Peak loads are met in message systems by building up queues which keep equipment utilisation high during lean periods such as lunch hours. There is no need to keep surplus capacity in equipment to ensure that a path can be established the moment someone dials a number.

Better utilisation of equipment in message systems extends also to terminals. A hardcopy terminal linked to a message system can serve ten to a hundred employees in an office: it is easy to distribute received messages; it is equally easy for outgoing messages to be queued up in a tray and then key-boarded by an operator. This sharing of terminal units has to be compared with sharing of telephones, which is quite problematic.

NATURE OF TRANSMISSION SYSTEMS NEEDED AND SIGNIFICANCE FOR MAINTENANCE

Compared to voice communication, message systems can work with lower signal-to-noise ratios. The use of error detection and automatic retransmission schemes in packet switching networks make them particularly tolerant to disturbances in signal paths. This fact, when combined with the earlier observations on low channel capacities means that simple equipment (eg. VHF transceivers) can be used satisfactorily to deal with rural communities too expensive to reach by standard microwave, UHF or cable systems.

A packet switching network used at the transport layer can be configured to have redundant links and to have some redundancy to handle node failures as well. Network software will take automatic action to bypass failed links and nodes, offering a reliable and effective answer to several types of maintenance problems.

It is worth repeating here the technically well-understood view that data communication does not require a perfect telephone network to be present as a prerequisite. Technological opportunites created by computer networks make it possible to utilise old, low-cost transmission systems for new applications, employing only a small fraction of their capacities for this purpose.

USER ORIENTED FACILITIES

The basic features to be considered are :

a) an integrated network accessible from privately owned or leased terminals, terminals shared by small user groups, from public terminals "rented" for individual sessions, and from public terminals operated by systems staff handling a telegraph-like service

b) use of messaging as the primary mode of communication, on-line dialogue being either unavailable or being a special, secondary mode of usage, avoiding line switching in either case

c) use of packet switching at the transportation level

d) use of packet broadcast channels where appropriate, as when a satellite channel is being used

e) built-in facilities for handling speed conversion and code conversion

f) interfaces to existing facilities such as Telex and telegraphy

g) facilities for transmission and typing out of correspondence quality text

h) for multicopy facilities based on mailing lists

i) for reliable authentication of origin

j) for system generated acknowledgements

k) for encryption

l) provision for a "subject" field, possibly a keyword field, and facilities for indexed archiving, and

m) mailboxes which can be accessed from any physical terminal by users authenticated appropriately.

Some of the facilities listed here need considerable examination before an implementation can be visualised. For instance, acknowledgements can be generated by the system to indicate:

a) a message has reached the mailbox concerned

b) recipient, authenticated by his password, saw indication of the message concerned in a summary display (listing origin and subject)

c) recipient saw the text of the message concerned (at a specified time, on a specified terminal)

The system could enter such information in a status table as a part of information being archived, to provide records of message delivery. In addition, senders of messages could ask for active notification to be generated automatically at a specified stage.

Facilities such as conferencing seem to be less important than the ones listed above for the specific purposes under discussion here. Provision for redirecting a message by an authorised and authenticated user seems necessary. Facilities for key-boarding and editing of messages are essential and are taken for granted.

For a computer based message system, many of the features listed here are available at low (incremental) costs. Therefore, despite their being fairly sophisticated features in a communication system, they seem very appropriate to the type of systems being visualised here. For instance, for business and government communication, the cost/benefit ratio of incorporating encryption seems to be very attractive.

EQUIPMENT AND COSTS

Using a $ 50,000 computer system and 50 terminals at $ 1,500 each, one can create a single node in a network of the type discussed above, providing for necessary redundancy and inter-node communication interfaces, including modems. One could use terminals with integral modems or with external modems. One could also consider using teleprinter interfaces with teleprinters performing the role of messaging terminals. Relatively fast external modems (eg. 4 wire, 1200 BPS modems) can be used with special high speed terminals (eg. 180 ch/second) to serve large groups of users.

At an approximate investment (computer, modems and terminals only) of $ 2,500 per terminal, a system like this can handle a large volume of messages. The terminals can print (at 30 ch/second) over a hundred messages, each a hundred words long, per hour. Two or three 4800 bits/second synchronous channels can handle the required volume at the inter-node level, taking into account facts such as terminals being tied up by (slow) key-boarding most of the time, and terminals doing output only for four or five hours per day. Counting interest on investment, depreciation and maintenance, annual costs on the computer system and terminals could be as high as 40% of the capital involved. Even at this cost ($ 50,000), the cost per message (considering only computer and terminal costs) turns out to be impressively low, in view of the fact that one can handle around 100,000 messages per year per terminal.

CONCLUSION

This limited exercise was meant to arrive at the definition of a class of message systems that could meet significant communication requirements in developing countries, without making undue assumptions about the resources or technological infrastructure available. If it persuades the reader to believe that thinking small is profitable, in the context of communication problems of large developing countries, this paper would have served a significant part of its purpose.

REFERENCES:

[1] Ramani, S., On Packet Switching for Communication Systems, Journal of Institution of Electronics and Telecom Engrs., 1978, Vol.24, Nos. 10 & 11, pp. 476-482.

VOICE MESSAGING

Chairman - D. Farber

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP,1981

A VOICE MESSAGE SYSTEM

Danny Cohen
USC / Information Sciences Institute
4676 Admiralty Way
Marina del Rey, California 90291

ABSTRACT/INTRODUCTION

Real time packet speech communication was developed and demonstrated across the ARPAnet several years ago. It is based on a high level Network Voice Protocol, NVP, which is capable of using any of several different speech encoding (vocoding) techniques, such as LPC, CVSD and PCM.

A Voice Message System, VMS, was also developed and demonstrated. This VMS was designed to operate over the ARPAnet. It is based on NVP and is available for transnet use by any NVP station.

This paper discusses the philosophy of the VMS, its architecture, its implementation and its connection to the ARPAnet text message system.

GENERAL

Online (real time) speech is the most natural mode of communication among people. It is now possible to use general purpose packet data networks for transnet real time speech due to recent developments in the theory of signal processing, in the hardware needed to implement it and in packet switching technology. This capability was demonstrated over the ARPAnet as early as 1974. References [1] through [9] describe experiments in packet speech and the issues involved, both for the ARPAnet and for the Ethernet. [10] discusses the NVP, the high level (Host/Host) protocol which was used over the ARPAnet.

Even though real time speech is by far the best mode of communication among people, it is not free of problems, especially in the office environment. These same problems are frequently evident in telephone communication.

The two main problems of real time speech communication are:

- The need for synchronization of the participants
- The lack of a permanent record whenever needed.

Synchronization of participants is the need for both parties to be at their phones at the same time. This synchronization, which is obviously necessary for interactive communication, may be bothersome ("Sorry, Ms. Jones is in a meeting now") or even impossible ("Sorry, she is on a trip to the Amazon").

Typically, office-mates and secretaries alleviate the difficulties associated with the people synchronization issue ("May I take a message?"). However, they can rarely provide a complete permanent record of a call unless it is extremely short.

One may notice that many phone calls do not really require synchronization and may be perfectly well handled by a secretary. These calls are typically those that could be replaced by a network message (aka "Electronic Mail", or EM). An example would be "I found the book in the library".

Telephone answering devices were designed to alleviate the above problems. They cannot totally solve these problems and they may not be as efficient as a good secretary, but they are always available and never forget to relay a message.

The main feature of these devices are:

- Access for recording from any telephone on the STN (the Switched Telephone Network) without any special equipment.
- Controlled access for playback from any phone on the STN without any special equipment.
- Provision of a permanent record of the call. Many of the commercial models though, restrict the length of the call and do not provide an easy way to "archive" recorded calls.

The VMS described in this paper has the above features, and more:

- It is based directly on NVP, hence it will always be compatible with any new vocoding technique that will be on the network.

- There is no need to have a separate VMS for each subscriber because different subscribers may share the same unit without compromising their privacy.

- The VMS can support multi-destination messages. Hence, there is never a need to record (and store) a message more than once, regardless of the number of its recipients.

- The VMS uses computer file- and archival-systems for keeping the recorded messages.

- Message integrity may be obtained for the VMS in several situations in which integrity is impossible to obtain for real time speech. For example, insufficient network bandwidth always damages the integrity of real-time speech, but not necessarily so in the case of off-line speech, as could be used by the VMS.

Most of the potential users of the VMS already use text Electronic Mail, EM, which in our case is the ARPAnet message system. Since the VMS may be implemented as a separate system, it contains the "hooks" that are needed for using the existing EM system to alert users about pending voice messages.

For situations where vocoders are not as readily available as text terminals, it is possible to use the text medium, such as the EM, to help users in communications about their voice messages, as will be described later. In addition, the sender of a voice message may associate with the message some fields of text such as "SUBJECT", for example. These text fields should be available for retrieval using text terminal. These fields may also be used to provide information about pending voice messages when alerting users via their EM.

On the other hand, in the dual situation where text terminals are not as handy as the voice system, the voice capabilities should be used not only for communication of the voice messages but also for communication about the text messages.

We have a system at ISI that allows one to call in from any touch-tone phone to inquire about his EM, in a controlled manner. The system talks to the caller, giving him details about the messages in his mailbox: the number of messages, as well as the sender, the subject and the length of each message. Words that are not known to the system are spelled out over the phone. Hence, a user who expects an important message may check for it from anywhere by using a public (touch-tone) phone, without the need to carry a text terminal (and modem) with him. The terminal would be required only to retrieve the entire message, once it had arrived.

Obviously, such a voice front-end could also be implemented directly on the VMS. However, if the voice front-end already exists for the EM, and if the VMS uses the EM as part of its own front-end (alerting, etc.) then the voice front-end capability is already available for the VMS, indirectly.

SYSTEM ISSUES

Digital voice encoding (vocoding) requires data rates between 1kbps and 64kbps during periods of voice activity. Only a trivial data rate is required for periods without voice activity, the "silent" periods.

A very small number of integrated circuits are required for the vocoding. Traditionally the higher the compression rate, the more complex the device. Discussion of voice encoding ("vocoding") and compression may be found in references [11], [12] and [13].

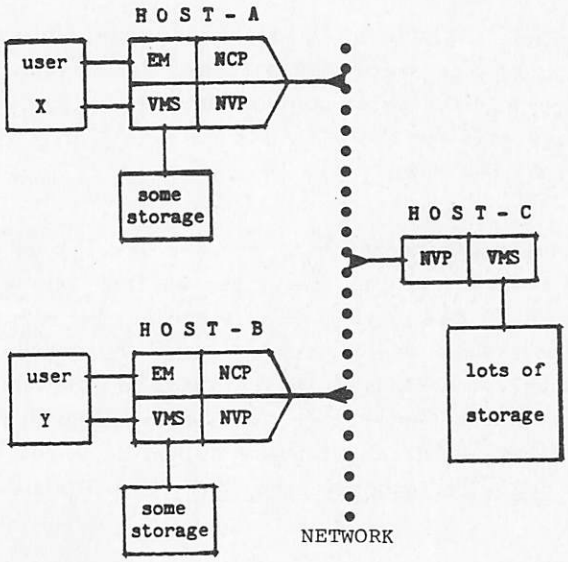
Today, only a few circuits are needed for PCM (64kbps) or for CVSD (8-32kbps). LPC synthesizers (2Kbps) are implemented on a single low cost circuit, and the analyzer is expected soon. Additional logic is also required for detecting silence and encoding it.

The motivation for speech compression is obvious: a 15 minute voice message at 64kbps requires 57.6Mbits, which is a (relatively) large storage requirement. Consider the American Declaration of Independence. It contains 300 words; when recited in 2 minutes it requires about 7.68Mbit of PCM, or only 0.288Mbit of LPC. The same declaration however requires less than 0.02Mbit for its ASCII coding.

Due to the huge quantities of storage space which may be required for storing voice messages, it may be desirable to separate the VMS from the mailbox that is used by the EM system. This is because many EM systems are implemented on facilities which have storage limitation which cannot support even few minutes of speech.

Network performance, such as a (temporary or permanent) lack of sufficient bandwidth may damage the fidelity of real time speech. However, offline voice messages should not suffer from this possible deficiency. If the communication medium can support reliable communication, such as by using TCP [14] or X.25/X.75 [15] and [16], then it should always be used by the VMS, in its offline mode.

Several communications scenarios are discussed next. The figure below may be helpful in following this discussion. The best way for user-X on host-A to send a voice message to user-Y on host-B is by dictating it first inside the A-system. Later, possibly after some editing, the message should be sent via a reliable file transfer protocol (FTP) to the B-system. The recipient Y would then be notified about this voice message, via the EM system, and at his convenience, would play it.



This scenario requires that both users have sufficient storage space for the entire message, that the sender is allowed to store a file of such size at the destination host, that both sites have a common FTP, and that both be online at the same time. In a scenario involving personal computers this is not always true.

Let's consider the case that neither A nor B is able to store an entire voice message. In this case X should use storage space which is at host-C, where the separate VMS is implemented. The online speech communication mechanism (NVP) should be used to get the speech from A to C. The EM would be used to tell Y that a voice-message was waiting for him at C, and how to access it. At his convenience, Y would play the voice-message by retrieving it from C, using the online speech communication mechanism (NVP) again.

These two transnet speech communications from A to C and from C to B are performed by using NVP. Since NVP is designed for online real time speech, it does not use conventional reliability techniques such as retransmission and sorting out-of-order packets. Therefore, the voice message is exposed twice to possible damage by the unreliability of the transport mechanism.

However, since the VMS at C does not have to support real time reproduction of the voice message when it arrives from A, C could re-order any packets which arrive out-of-order. Retransmission of lost messages does not occur, since A lacks this capability which is never required for NVP.

NVP uses time stamps on each of its packets. C should never change them, so B always gets the timestamps as they were originally put on by A. Hence, if the A-to-C communication is bursty with a large delay variance, the final playback at B is not affected by it at all. Obviously, there is no way to hide the communication problems which occur on the C-to-B path, except by using a high value of the voluntary delay, D. This voluntary delay, D, which is also known as reassembly delay or reconstitution delay, is discussed in [5].

THE NVP COMPATIBILITY

The Network Voice Protocol, NVP, is defined in [17] and is described in brief in [10].

NVP has two major components, a control protocol and a data protocol. The control protocol is responsible for the call-establishment procedure, for call-monitoring and termination, and for achieving compatibility of the data protocol. The data protocol is the actual encoding of the speech according to a mutually agreed upon algorithm and its associate parameters. In addition, the data protocol includes silence handling and time-stamping.

The control protocol establishes the agreements that are necessary for compatibility by a process of negotiation. In this process one party suggests ways for doing certain things (e.g. "use LPC vocoding") and the other accepts or rejects these suggestions. This negotiation process lasts until either a complete agreement has been reached, or one of the parties decides to give up.

This negotiation process cannot always make any two arbitrary systems compatible with each other. However, it is designed to discover which combination of the parameters that are available to both parties yields the desired compatibility.

When the VMS is used the two parties cannot conduct interactive online negotiation between themselves, as described above. Instead, the VMS conducts separate negotiations with the recording party, the sender, and with the retrieving party, the receiver. Obviously, these two negotiation processes cannot be conducted simultaneously.

First, the VMS negotiates with the sender. In this negotiation the VMS is most agreeable and accepts all the suggestions put forth by the sender. The VMS accepts them and records them. Later, when the voice message is about to be retrieved, the VMS negotiates with the receiver. This time the VMS insists that all the parameters that were used before are used again. The VMS retrieves these parameters (that were recorded earlier) and is most persistent that they are used regardless of any suggestions made by the receiver. Obviously,

if the receiver cannot handle these parameters, the recorded voice message is useless, just as if it were recorded in a language that the receiver did not understand.

The VMS does not try to resolve incompatibilities between the vocoders of senders and receivers, just as telephone answering devices do not provide language translation.

ACCESS CONTROL

Controlled access for retrieving recorded messages is an essential part of any VMS. Even though similar controlled access is also an essential component of file-systems in shared environments, such as time-sharing computers, the VMS implementation requires more general features than those provided by computer systems for their files.

Our VMS provides two types of privacy measures for recorded messages: passwords and encryption keys. In addition to the passwords which every user has in order to gain access to his "mail-box" (i.e., to the messages which were recorded for him), users may associate individual passwords with each recorded message. The VMS would never allow access to a message for which a password was specified unless the password was included in the retrieval request.

The individual passwords are particularly useful for messages which are directed to a large set of possible recipients, not all of which are necessarily known explicitly to the sender.

The other protection measure is encryption. Since the VMS never interprets the voice data of the messages, any voice message could be encrypted in any way which is desired by the users. Note that the vocoding is already a coding process with known parameters. Encryption is nothing but modifying this coding process with some parameters which are kept unadvertised. As a matter of fact, note that there is no significant difference between the use of encryption and using some obscure foreign language when it is not expected.

Both types of privacy measures require some out-of-band communication between the end-users for communication of the password or the encryption key. There are many ways to achieve this

out-of-band secure communication, ranging from the usage of fixed-agreed upon patterns to time-dependent patterns and to public-key methods.

The former measure, the individual password, depends on the ability of the VMS to provide the desired protection. It requires the least effort on behalf of the end-users.

On the other hand, the latter measure, encryption, relieves the VMS from any effort at the expense of the end users which must be able to implement all the privacy measures by themselves, entirely on top (i.e., outside) of the VMS.

VOICE MESSAGES AND TEXT MESSAGES

Users typically prefer to have only one message system, both for voice and for text messages. Unfortunately, different devices are needed to handle each of these modes of communication, namely an audio device, like a telephone, and a character-oriented device, like an alphanumeric terminal.

Clearly it becomes feasible to integrate work-stations to include both voice and alphanumeric capabilities into a single integrated station, but unfortunately most users still cannot afford this luxury.

In spite of the differences between voice and text, the user would like to conduct his main interaction with one system, and to use the other only when needed. As long as text-to-voice and voice-to-text systems are not widely available, the different modes may be used only to find out about each other. For example, phones may be used to discover the existence of incoming text messages, and text-terminals may be used for interrogation of incoming voice messages.

In the ideal system voice and text should be totally interchangeable. Messages should be entered and retrieved in whichever mode happens to be the most convenient at the time. Unfortunately the technology which is needed to support this interchangeability is not available yet.

At present it is possible to build integrated systems in which text can be attached to voice messages and vice versa.

People who prefer to use telephones would like to be able to use them to find out about their incoming text messages, who sent them, when they arrived, what is their subject, their classification, etc. Likewise, people who prefer to use text terminals are interested in similar information about their incoming voice messages.

Note that the above is not a classification of users by their preference, but rather an indication of their preference which may depend on their location at any given time. For example, even a text-oriented person may prefer to use an available phone from an airport lounge, rather than wait to get to a text terminal.

In order to implement the mutual support of the voice and the text message systems, they must know about each other, and conventions (protocols) must be established. It turns out that a simple scheme in which each system prepares a short message in the other mode, containing a notice (summary) of its own messages, goes a long way for using each system as a front end for the other.

Another approach to the same problem is to provide each system with an interface to the other mode, e.g., a voice interface to text-EM and vice versa.

We have experimented with both approaches (voice interface to text-EM and text-EM interface to VMS) and found that both have the expected advantages.

Please note that the above discussion assumes a distributed text-EM system, but a centralized VMS. Obviously there is no magic about this choice. Each system could be implemented within a wide spectrum of distribution techniques, from a tightly centralized approach to a totally distributed system. We chose to discuss the VMS and the EM as centralized and distributed, respectively, not only because this was our choice of implementation, but also to demonstrate that there is no need for a unified approach to both systems Either system could operate without requiring that all of its participants use the same level of centralization/distribution.

The issue of using a local network in conjunction with a VMS is an independent question of efficiency. Both text and voice could be distributed at each site in any of several ways, from communication lines arranged in a star configuration to local networks of any

variety. Furthermore, there is no need for both the VMS and the EM to use the same local distribution model. Integrated local networks are very nice, but not an absolute necessity for the integrated operation of the VMS and EM.

SUMMARY AND CONCLUSIONS

Since the packet speech technology uses computers to handle real-time speech communication, it is possible to use their general purpose power to add speech storage capability. Once having this capability it is possible to construct on top of it VMS's which provide general-purpose computerized high-level user interfaces which are limited only by the imagination of their designers.

Generally speaking, it is possible to extend the notion of EM to include the proper handling of voice-messages in such a way that takes advantage of all the lessons learned during our long experience with various EM systems.

ACKNOWLEDGMENTS

Steve Casner, of ISI, helped both the writing of this note and the implementation of the VMS system at ISI. Both John Shoch of Xerox-PARC, and Jon Postel of ISI, also provided much help in the preparation of this note.

REFERENCES

[1] Markel, J.D., "Highlights of a Group Effort in Algorithmic Development for Packet Switched Voice Networks", Proceedings of the NCC, May 1981.

[2] Blankenship, P., et al. "Hardware Modularity for Packet Voice Terminals", Proceedings of the NCC, May 1981.

[3] Cohen, D., "Packet Communication of Online Speech", Proceedings of the NCC, May 1981.

[4] Shoch, J.F., "Carrying voice traffic through an Ethernet local network -- a general overview," Xerox Parc Technical Report, Aug. 1980. To appear in the Proceedings of the IFIP WG 6.4 International Workshop on Local-area Computer Networks, Zurich, August 1980.

[5] Cohen, D., "Issues in Transnet Packetized Voice Communication", Proceedings of the 5th Data Communication Symposium, September 1977, pp. 6/10-13.

[6] Forgie, J.W., "Speech Transmission in Packet-Switched Store-and- Forward Networks", Proceedings of the NCC, 1975, pp. 137-142.

[7] Forgie, J., and Nemeth, A., "An Efficient Packetized Voice/Data Network Using Statistical Flow Control", Proceedings of the ICC, Chicago, June 1977.

[8] Ross, M.J., and Sidlo, C.M., "Approaches to the Integration of Voice and Data Telecommunication", Proceedings of the NTC, Washington, D.C., November 1979.

[9] Ross, M., et al., "Design Approaches and Performance Criteria for Voice/Data Switching", Proceedings of the IEEE, September 1977.

[10] Cohen, D., "A Protocol for Packet-Switching Voice Communication" Computer Networks, September 1978.

[11] Markel, J.D., and Gray, A.H., Jr., Linear Prediction of Speech, New York, Springer-Verlag, 1976.

[12] Flanagan, J.L., Speech Analysis, Synthesis, and Prediction, New York, Springer-Verlag, 1972.

[13] Rabiner, L.R., and Schafer, R.W., Digital Processing of Speech Signals, Englewood Cliffs, N.J., Prentice-Hall, 1978.

[14] Postel, J.B., ed., "DOD Standard Transmission Control Protocol", IEN 129, RFC 761, USC/Information Sciences Institute, NTIS ADA082609, January 1980. Appears in Computer Communication Review, Special Interest Group on Data Communications, ACM, 10:4, October 1980.

[15] Rybczynski, A.M., et al., "A New Communication Protocol for Accessing Data Networks - the International Packet Mode Interface" Proceedings of the NCC, 1976, pp. 477-482.

[16] Rybczynski, A.M., et al., "Design of the Datapac X.75 Internetworking Capability", Proceedings of the Fifth International Conference on Computer Communication (ICCC-5), October 1980, pp. 735-740.

[17] Cohen, D., "Specifications for the Network Voice Protocol (NVP)", USC/Information Sciences Institute RR-75-39, March 1976, and as RFC 741, NIC 42444, 22 November 1977. Also in [18].

[18] Feinler, E., and Postel, J.B., eds., "ARPANET Protocol Handbook", NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

AN EXPERIMENTAL VOICE-MESSAGE SYSTEM CONTROLLED BY WORD-RECOGNITION

P. Mermelstein and V. Gupta

Bell-Northern Research
3 Place du Commerce
Nuns' Island, Quebec H3E 1H6
Canada

We discuss considerations in the design of an experimental voice message system controlled by word recognition. The feasibility of speaker-independent recognition of telephone-transmitted digits and yes/no/stop control words is considered for a wide variety of speakers and accent groups. The results of our simulations suggest that systems be controllable by both multi-frequency signalling (DTMF) and word recognition. The experimental voice message system with word recognition will be demonstrated as part of the presentation of this paper.

INTRODUCTION

While computer-mediated text messaging systems are available today in many large corporations, voice messaging systems are truly in their infancy. Experimental systems are in use at IBM at Yorktown Heights, by the 3M Company in St. Paul and a few other locations. Additionally, a field trial is under way by the Bell Telephone Co. of Pennsylvania in Philadelphia. Before discussing an actual experimental voice messaging system, let us focus on the issues that have allowed text messaging to get off to a relatively fast start while voice messaging has developed much more slowly.

The principal economic factor standing in the way of voice messaging has been that of storage costs. A text that takes 10 seconds to speak requires roughly 2000 bits for its storage, whereas standard PCM coding of the corresponding speech would require 640,000 bits, resulting in a difference factor of 320. Efficient coding techniques can reduce this by a factor of four without degrading speech quality. Against this we must weigh the main impediment to text messaging, namely the requirement for a keyboard-equipped terminal, as opposed to the plain old telephone that can be used for voice messaging. However, direct comparison of the requirements is appropriate only if these services are competitive and not complementary. Let us examine in further depth the occasions when one might prefer to use one or the other form of messaging.

Leaving a voice message is currently most widespread in the case of a noncompleted telephone call - you wanted to speak to a person but he was busy or otherwise unavailable. We may imagine other situations where simple information must be transmitted one-way only, such as confirmation of receipt of a document. Where simultaneous participation of the two parties to the communication is really not required, messages may in fact be more efficient than telephone conversations. Voice messaging can be considered a replacement for a human attendant who would otherwise be required to transcribe the message, or for the simple analogue recording devices that can be connected to your telephone. In contrast, text messaging is particularly advantageous when the same information must be transmitted to many people, such as announcement of a meeting. Also, if the text

already is present in a computer, its transmission by a message is particularly simple. Thus text and voice messages are most convenient to use in different circumstances and appear primarily complementary from the service point of view. Since each can also satisfy the function of the other, albeit less conveniently, only in the secondary case do they appear competitive. If a user is constrained to use only one messaging service, he may pick either voice or text depending on his personal or work requirements. If both are available, he will likely use them selectively.

VOICE MESSAGING

The main advantage to voice messaging, as discussed above, is its potential availability from any telephone terminal. However, messages do not consist of text alone, but carry control information as to source, destination, priority, etc. Such information is most conveniently entered in a conversational mode in response to prompts from the message system. For voice messaging, verbal prompts are most appropriate, and response may be either by speech or by some secondary signalling method such as dual-tone multi-frequency signalling (DTMF). The advantage of the first is its convenience; control information and text are input using the same modality. Its disadvantage is the limited performance of current speaker-independent word recognition systems. DTMF signalling is faster and more accurate, however, the limited penetration of tone-transmitting telephones violates the near-universal availability that motivates introduction of voice messaging in the first place. We have conducted experiments with systems controlled by both techniques and feel that systems that can be controlled in either mode will find best initial acceptance. Since entry of control information by means of DTMF input is practically self-explanatory, we will only discuss in detail our experience with word-recognition.

WORD-RECOGNITION CONTROL

To maximize recognition accuracy with currently available word recognition techniques, the designer of the control protocol must limit the potential inputs from the sender or retriever of a message to a choice between very few alternatives. In most cases a "yes" or "no" input suffices. Only to identify the sender and receiver must we allow a wider variety of inputs, and the best combination between simplicity of use and good recognition appears to be a multi-digit code. To maintain parallelism with touch-tone controlled system, the same user codes are desirable. To enhance the accuracy of the word recognition input addition of a check digit appears sufficient. With the aid of a check digit most misrecognized codes can be rejected as unacceptable and input is slowed down only slightly. For those users whose speech characteristics force their input to be more frequently misrecognized, a digit by digit verification technique can be used which enhances accuracy while significantly slowing down the input rate. With this technique, each digit is played out by the system as recognized and confirmed by the user with a "yes"/"no" response. A "no" confirmation is followed by a playout of the second choice from the digit recognition and the procedure is continued until a "yes" confirmation is received. To maximize the acceptability of such a service, operator assistance should be provided for the few speakers who encounter repeated difficulty in using the system.

Other branch-points in the protocol may call for a response of "insert" or "recall" of messages, "yes" and "no" to a query for additional inputs, requests for playback and possible revisions of the message. The selection of the words to be used in responding to the system should be guided by human factors considerations as well as the need to maximize the phonetic differences and thus minimize the possibility of misrecognition.

WORD RECOGNITION PERFORMANCE

The message system has been tested using an experimental word recognizer designed to operate with speech input over the telephone lines and without regard to the speaker. The recognizer is trained by collecting telephone-transmitted productions of the individual words from large numbers of speakers. A clustering procedure is used to separate the productions of a word in the training set that are sufficiently similar that a single reference form suffices for their representations. We are currently using ten reference forms per word and each test word is compared to all reference forms of all words for maximum similarity. Comparison takes place using a representation of speech energy in frequency-time space where the words are aligned for maximal similarity in time. Simulating the process with a high-speed array processor (AP-120B) allows near real-time (2 seconds maximum) recognition response. With special-purpose hardware under construction, real-time response to one user appears easily attainable.

Experimental results with a 20-word vocabulary for native speakers indicate 97% correct recognition of individual words. For French-Canadian speakers speaking English, performance is somewhat lower, namely 93% [1]. By rejecting inputs that appear ambiguous to the machine, the frequency of misrecognition can be significantly reduced. Furthermore, multi-digit user codes can be selected so as to include check digits for error protection thus further minimizing the probability of misidentifying the sender or recipient of a message. The system has now been tested over a variety of local and toll telephone connections. Performance is somewhat sensitive to line noise and variations in the severity with which the high-frequency speech components are attenuated on toll connections. Techniques for noise cancellation and line equalization are currently being explored.

TECHNICAL CONSIDERATIONS IN STORING OF VOICE MESSAGES

To minimize the storage requirements, the speech signal constituting the message text should be encoded as efficiently as possible while maintaining acceptable reproduction quality. However, the complexity of the most efficient coders, such as, for example, adaptive predictive coding at 16 kb/s, may impose a coding cost in excess of the storage savings achieved. To minimize the overall costs of such a system, simple waveform coders such as adaptive-differential PCM (ADPCM) or continuously variable delta-modulation (CVSD) are preferable [2]. The system to be demonstrated uses ADPCM at 24 kb/s for storage of both data and prompt messages.

Large moving-head disks appear to offer the least storage costs for sizeable message libraries while at the same time providing adequate speed of retrieval. As shown in Fig. 1, significant additional cost is represented by the communications-controller with high-speed buffer store that matches one high-speed single-port disk channel to the relatively low-speed but real-time requirements of the actual speech communication channels. Separate codecs or code converters, one per input channel, allow a reduction in the buffer memory of the communications controller. The code-conversion is not executed during the transfer of data to and from the mass store, despite the fact that then only one code-converter would be required, because of the excessive speed requirements imposed by the high data rate. Additionally, the code converters monitor the speech/silence condition of the input channel. Silence intervals (background noise) prior to the start of the message and after the end of the message are deleted. Silence intervals longer than 250 ms within the message are reduced to 250 ms.

GENERATION OF PROMPT MESSAGES AND TIME OF RECEIPT INFORMATION

Prompt messages appear necessary to guide the user through the protocol for sending or retrieving messages. The conversation is controlled by the message

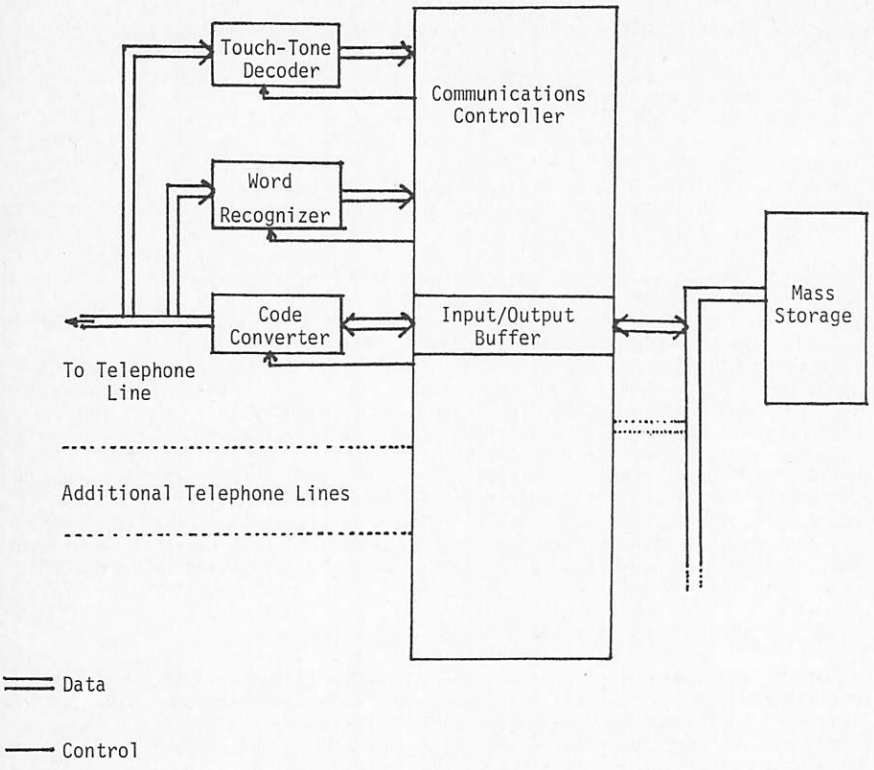


Figure 1. Simplified block diagram of experimental voice-message system

system, at each branch-point in the flow of the conversation, information is requested from the user and a time window is opened for the receipt of a response. Prompt messages are recorded off-line, digitized and encoded into special prompt files. Message assembly is required only for generation of identification information e.g., "message received on twelfth April nineteen eighty one at ten thirty a.m.". Words and phrases, as underlined, are recorded independently. Words corresponding to the actual date and time of receipt are substituted into the appropriate sequence. As long as the words are spoken with a reasonably neutral intonation, concatenating the words appears acceptable to label the messages. For somewhat improved quality, different inflections could be used depending on the position of the word in a phrase, for example, for "ten" of "ten-thirty" and for that of "four-ten".

The acceptability of concatenated labels appears to be related to the fact that they carry only auxiliary information accompanying the actual message. It is unlikely that the same technique would be acceptable to generate messages from a text form because the intelligibility requirements on the contents of the message are much more severe.

SYSTEM CAPABILITIES

The message system currently allows insertion and retrieval of messages. To input a message, the user gives his identification code and that of the intended recipient. After recording his message, the sender may listen to it and re-record it if desired. He can also insert additional messages without redialling the system by just giving additional destination identifications. Although broadcast of messages to a defined list of addressees is clearly possible, it has not yet been implemented.

To retrieve a message, the user gives his identification code and is told whether there are any messages on file for him. They can be retrieved selectively in any order. To encourage users to delete messages as soon as they have been received, users are asked whether they want their messages saved.

No provisions have yet been made for automatic delivery of messages, although that appears a clearly desirable feature. Delivery techniques need careful study to ensure that delivery is made to the intended recipient and not to whoever happens to answer the telephone. Preventing delivery of messages in situations where the calls are forwarded to message system poses an additional challenge. Clearly if a message system delivers a message to another system, it can deliver it to itself, thereby quickly exhausting available storage space. Thus messages should only be delivered to people.

The experimental system uses 14 Mbytes of a 300 Mbyte disk. 170 users can share a total speech storage duration of 70 minutes. A specially designed file system assigns storage space on demand and releases that space to the available pool once it is no longer required. The system is limited to communicating with one user at a time.

In any practical system, the number of users that can be served very much depends on the length of messages and their average residence time in the system. The dearth of data in this regard suggests that initial systems be designed to be modular and easily extendable if initial assumptions are not borne out by experience with practical users.

DISCUSSION

Voice messaging with word-recognition control is technically feasible. It can provide a useful extension of messaging services that would otherwise be accessible

only from DTMF equipped telephone sets. Further studies on user acceptance of the speech-recognition features are needed before the system can be said to be truly universally accessible. In particular, voices of children and old people have not yet been included in the performance studies.

Voice messaging appears particularly useful when unstructured messages need to be transmitted. Selective editing of voice messages is feasible but appears clumsy. For short messages, repetition of the entire message is quicker and simpler to implement.

As voice messaging systems develop, both on private and public networks, interconnection and standardization problems will have to be faced. Despite the increased transmission costs, because storage at different centres may be carried out with different coding techniques, messages probably will have to be converted to 64 kb/s PCM for transmission to other message centers. Packet networks appear particularly suitable to handle message traffic between message centres because the real-time transmission requirements disappear.

Voice messaging may not be ready for effective standardization activities because of the lack of experience we have with such systems. However, the earlier discussions are started on this front, the better the chances for avoiding the development of incompatible systems.

It appears too early to address the question of the cost-effectiveness of word-recognition control as opposed to DTMF control of message systems. Where DTMF facilities exist, DTMF decoding is clearly less expensive. As application of word recognizers becomes more widespread, their cost is likely to be reduced significantly. It is in the large centralized systems, where the recognition cost can be shared by a large number of users, that speaker-independent recognizers will find their initial application.

REFERENCES

[1] V. Gupta and P. Mermelstein, "Effects of accent on the performance of an isolated-word recognizer," J. Acoust. Soc. Am., to be published.

[2] N.S. Jayant, "Digital Coding of Speech Waveforms: PCM, DPCM, and DM Quantizers," Proc. IEEE, 62, 611-632, 1974.

FACSIMILE MESSAGING AND EXTENSIONS

Chairman - K. Ono

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

NETWORK ARCHITECTURE OF ADVANCED FACSIMILE INTELLIGENT COMMUNICATION SYSTEM

K.Watanabe, Y.Kirihara, T.Endo and T.Kamae

Yokosuka Electrical Communication Laboratory
Nippon Telegraph and Telephone Public Corporation
Yokosuka, Kanagawa
Japan

The outline of an advanced facsimile intelligent communication system FICS is presented. FICS consists of the simplifie d terminals and, the economical and intelligent network which takes advantages of digital transmission, conversion and storage. The advantage of a message switching system as the upper portion network of FICS is derived through the investigation of the holding time, network transfer gain and cost/performance.

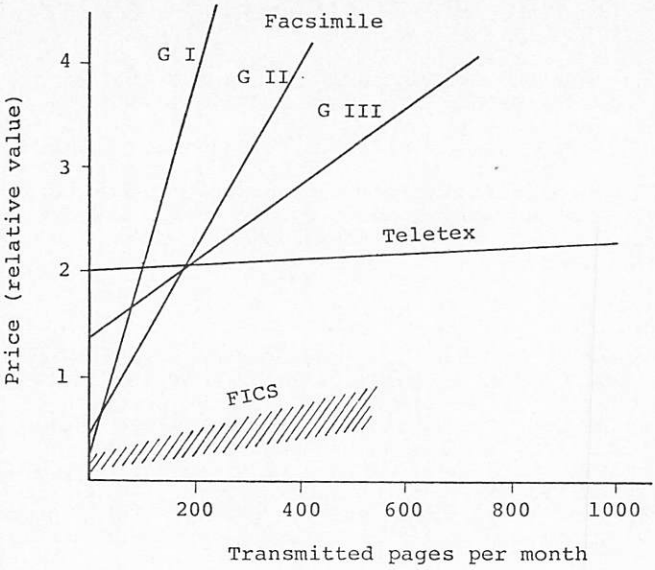
1. Introduction

Recently, facsimile communication attracts attention as a promising post-telephone service for the reasons that
(a) facsimile can provide the receiver-oriented communication by implementing an automatic receiving function,
(b) facsimile can transmit any kind of characters and graphics,
(c) handling is simple,
(d) facsimile communication can be economical even for low traffic users as compared with other record communication media as shown in Fig.1, and that
(e) facsimile sets can be used as data communication terminals through facsimile data converter.[1]

The present facsimile communication is executed through the general switched telephone network (GSTN) and many enterprises take advantages of this record communication. However, the usage is limited to enterprises since both terminal and transmission costs are high at present. Nippon Telegraph and Telephone Public Corporation (NTT) is now developing the Facsimile Intelligent Communication System (FICS) which can provide economical and multipurpose facsimile communication services for small shopkeepers, home industries and even the general public as well as enterprises. The first version system (FICS-1) is expected to start the commercial service in early 1981.[2] Research and development have been continued on this facsimile communication system to improve the cost and user convenience. The second version of the facsimile communication system is called FICS-2.

2. Service Outline

NTT will provide subscribers with facsimile communication network service and terminals. The outline of FICS-2 service is as follows.
(1) Facsimile terminals
FICS-2 adopts two types of facsimile terminals MF-1 and MF-2, both of which are designed and developed for the system. Specifications of MF-1 and MF-2 are described in Table 1.



Evaluation bases
Document size : A4
G-III coded picture : 35KB
Teletex code : 3.2 KB
Communication distance : 100km

Fig.1 Examples of record communication cost

Table 1 Facsimile Terminal Specification

Kind of terminal / Items	MF-1	MF-2
Document Size	A5 (149mm × 210mm)	A4 (210mm × 297mm)
Transmission	Analoque (AM-PM-VSB)	Analoque (8ph PSK V27ter MODEM)
Redundancy Reduction	All white line skip method	Modified Huffman coding
Transmission Procedure	Simplified T-30 tonal	T-30 binary
Procedure Time	12 seconds	10 seconds
Transmission Time	79 seconds	71 seconds

(2) Basic service functions
Regardless of the kind of facsimile calls, the following basic service functions are equipped in FICS-2.
(a) Appending of the originator's telephone number and the time of the call origination to each delivered document.
(b) Repeated access trial to a receiver : In the case that the originated call encounters the circuit busy, FICS repeats the connection to the receiver at the predetermined intervals.
(c) Automatic reception without ringing.
(d) Return of undeliverable documents.
(e) Parameter conversion between different types of terminals.
(3) Communication services
The following services seem to be useful :
(a) Basic calling (End-to-end connection)
(b) Multi-address calling
(c) Confidential delivery
(d) Facsimile box : FICS stores incoming documents and delivers them on the receiver's request
(e) Urgent delivery
(f) Computer access through a media converter

The access to FICS and the choice of services are executed by user's dialing. The numbering plan for FICS's services is shown in Table 2.

3. Network Design

3.1 Design Objectives

(1) FICS is intended to economically accomodate low traffic users as well as high traffic users. The cost of the subscriber loop is usually the most dominant factor, and thus must be kept as low as possible especially for low traffic users. FICS shares the lower portion with the general switched telephone network for economical efficiency and construction easiness.
(2) The upper portion of FICS must fit the facsimile communication characteristics to make facsimile transmission cost as low as posible under the condition of unified network configuration without distinction of facsimile terminal types. The digitalized network suits the upper portion, since a digital system is essentially familiar to facsimile signal consisting of two levels.
(3) An asynchronous half-duplex transmission control procedure is introduced in MF-1 and MF-2 terminals mainly for the reasons of terminal cost saving.

Table 2 Numbering Plan for FICS's Service

Communication Prefix	Service Prefix	Destination Number
1 X_i Y_i ($^{161}_{162}$)	1 k_i l_i (m_i) # k_i l_i (m_i)	National Number(s), xx---x(*xx---x*xx---) Abbreviated code (s), *ΔΔ(*ΔΔ*Δ--) Password Number, □□□□

3.2 Choice of the digitalized upper network
(1) Holding time
Circuit, message and packet switching (CS,MS and PS) systems are considered as candidates for the digitalized upper network. Although their strict definitions are very difficult, they can be compared as shown in Table 3. The FICS network can be illustrated as in Fig. 2. Examples of typical end-to-end facsimile communication call procedures for three types of upper networks are shown in Fig. 3. This figure shows that

Table 3 Classification of the Digital Switching Network

System / Item	Circuit switching	Message switching	Packet switching
Switched unit	call	message	packet
Information path	both ways	single way	single way
Switching signal channel	out band (common signal channel)	out band	in band

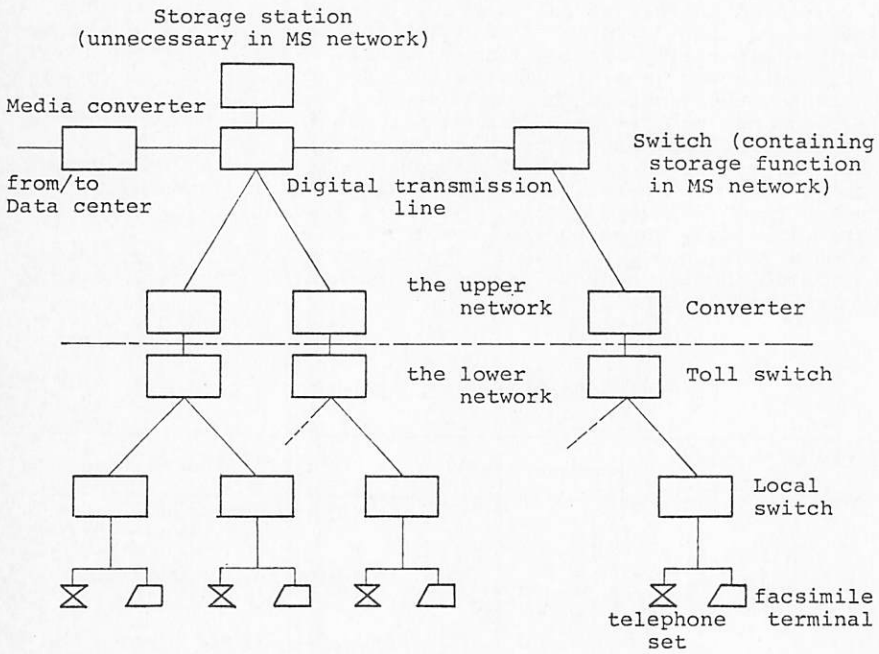


Fig.2 FICS network configuration

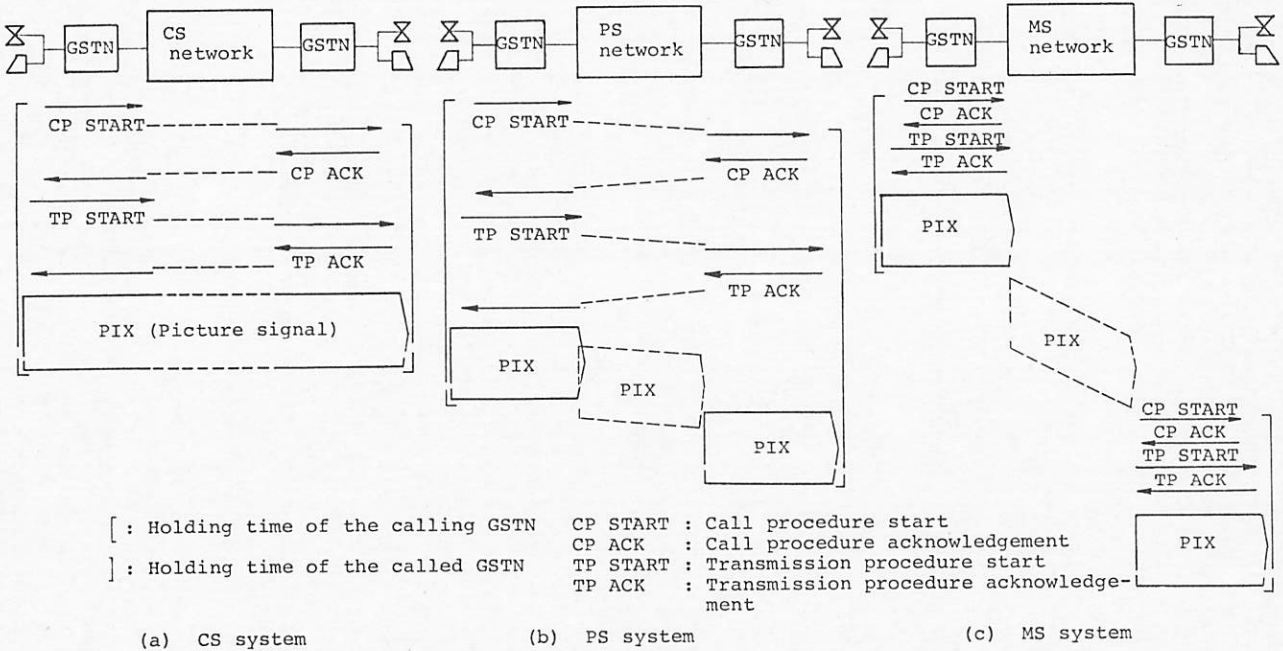


Fig.3 Examples of end-to-end facsimile communication call procedure

(a) in CS system, both the lower and the upper networks are occupied simultaneously for longer term than in MS system, that
(b) in PS system, the lower network and the converters are occupied for much longer term than in MS system due to the packet transfer delay in the upper network, and that
(c) in MS system, although end-to-end message transfer delay is larger than in the others due to message storage in the switches, the lower network and the converters are occupied for much shorter term than in the others.

(2) Network transfer gain

Converters and switches have the following effects on the facsimile signals transferred from end-to-end.

(a) Converter effects
Digital multiplexing can be adopted for transmission in all systems. In MS and PS system, transmission control signals can be compressed. In CS system, compression in the converters is meaningless since the transmission circuit is held always throughout one call.
(b) Switch effects
In MS system, transmission efficiency can be increased by the traffic quereing effect due to storage.

These effects can be quantified by defining the network transfer gain as the ratio of the "transfer capacity" (the number of documents to be transferred through a voice channel or a 64 Kb/s digital channel) before converters or switches to the one after them. The network transfer gain and transfer capacity of each system are illustrated in Fig. 4, where the transfer capacities are normalized by the capacity in the lower network of MS system. This figure shows that the MS system is superior from the point of the network transfer gain and the transfer capacity to the other two.

(3) Cost/performance of the network

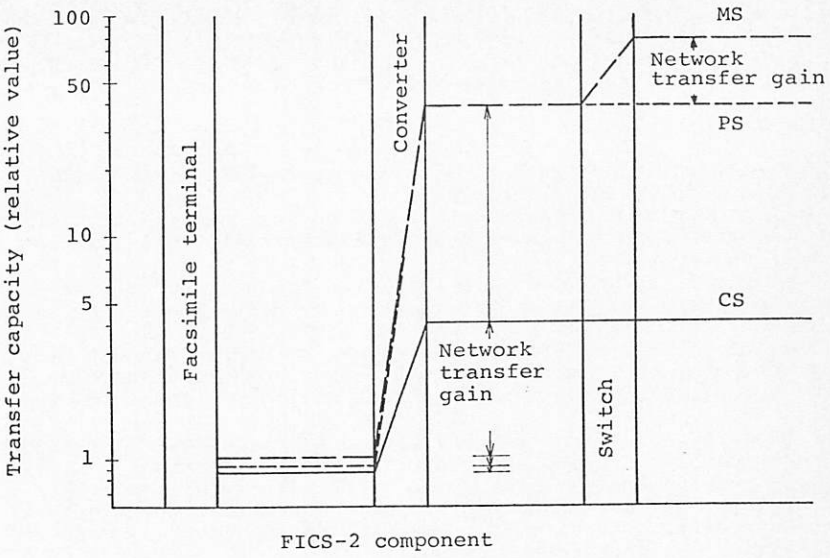
An increase of the network transfer gain reduces the lower network cost and the transmission line cost, but raises the processing cost. Therefore, the network configuration has to be evaluated from the "cost/performance" viewpoint. The network cost/performance is equivalent to the cost per document. The cost/performance comparison among three types of networks is done on the following bases,

(a) Evaluation criterion : Call-through cost per document
(b) Kind of call : Basic end-to-end call including the case temporarily stored in the storage station when circuits are busy in the called side lower network.
(c) The network model : As shown in Fig. 2, the lower network is the existing GSTN within a toll area, the converters are located in the toll center, and switches and storage stations are located in the district center or the regional center.
(d) The traffic model :

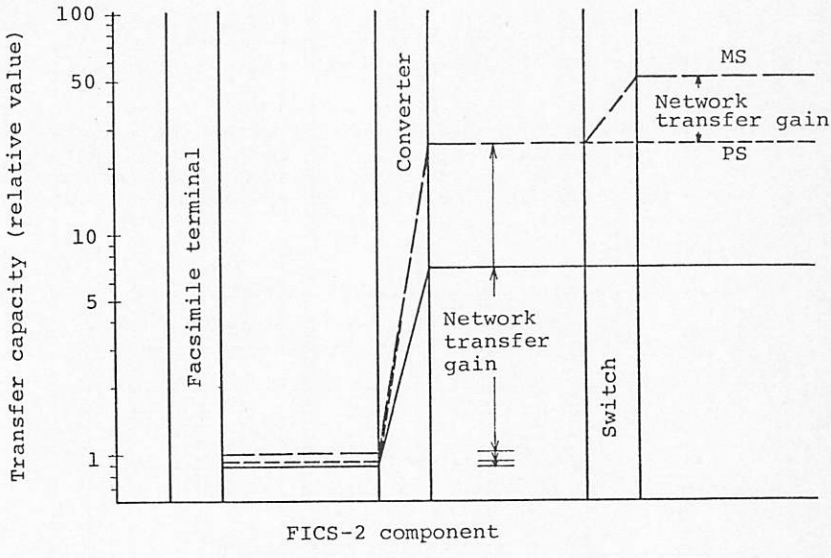
$$\frac{\text{intra-switch call}}{\text{inter-switch call}} = \frac{6}{1}$$

The evaluated result is illustrated in Fig. 5. The followings are estimated from this evaluation.

(a) MS system is superior in the cost/performance to the other two, which do not need storage in the end-to-end call.
(b) In CS and PS system, the increase of cost due to the increment of holding time of the lower GSTN is not negligible.
(c) In PS system, the use of short size packet, e.g. 256 bytes, causes the high switching cost. However, the longer packet size causes the lower network cost to become higher due to the increase of holding time. In conclusion, PS system is slightly more expensive than MS system; furthermore, MS system can easily and



(a) MF-1



(b) MF-2

Fig.4 Network transfer gain and transfer capacity

less expensively provide the storage services such as confidential delivery and facsimile box services without extra facilities, since the switches contain storage function. As a whole, MS system seems best fit for the upper network of FICS.

4. Function distribution

In FICS network, functions are distributed to the network components based on the following principles.
(1) Only indispensable functions should be assigned to local switches (LS) and other functions should be concentrated to toll switches (TS) in the lower network.
(2) The interfaces to terminals should be concentrated to the converters to enable new types of terminals to be connected without much alteration of the switch.
(3) The interfaces to data centers should be concentrated to the media converters to enable various types of centers to be connected in the future. FICS-2 assigns various functions in the following ways :

(a) LS : FICS's communication prefix and originators' telephone number detection, no-ringing trunk circuit selection
(b) TS : service prefix detection, subscriber confirmation, charging control and administration, connection route control of the upper network, circuit connecting of the lower network and facsimile calling signal transmission
(c) Converter : transmission procedure control with terminals, originators' number and origination time appending, coding and decoding
(d) Switch : link connecting of the upper network, picture signal storage, parameter conversion, storage service control
(e) Media converter : facsimile-data conversion,[1] computer connection control.

In FICS-2, the converter, the switch and the media converter are realized as Facsimile signal conversion and procedure control equipment (FCAP), Facsimile storage and conversion equipment (STOC), and Facsimile data conversion and interface control equipment (FDIC), respectively.

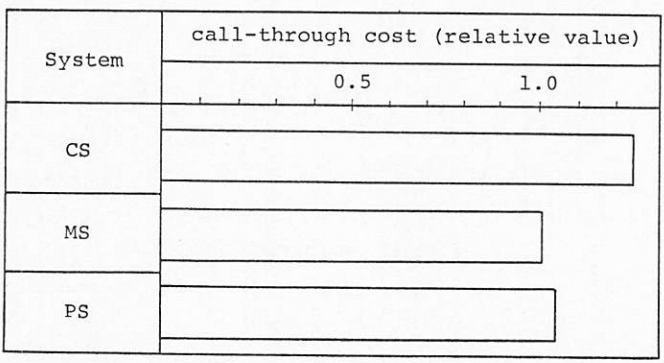


Fig.5 FICS's call-through cost

5. Network operation outline

Examples of typical end-to-end and center-to-end facsimile communication call procedures in FICS-2 are shown in Fig.6 and 7. A calling party dials a particular prefix number "161" and a destination telephone number. The local switch (LS) receives the number and sends it to TS, together with the originator's telephone number. Each TS keeps the information on the facsimile subscribers in its subscriber file. The TS checks whether the originator is a subscriber or not, and sends the destination number to the destination TS through the common signaling channel to ask whether the called party is qualified for facsimile communication or not. Then, the calling side TS requests STOC to receive a facsimile call. STOC orders FCAP to start the receiving operation, FCAP receives, digitalizes and packetizes the facsimile signal. In this digitalizing process, FCAP-I encodes the picture signal into the modified Huffman run-length code. The coded signal is transmitted in the packet multiplexed form to STOC through a 64 Kb/s digital line. STOC stores the coded signal in the picture signal file disk memory (DKM) and passes it to the called side STOC through a 64 Kb/s line after receiving a full page of coded picture signal. The call control information is transmitted through the common signalling channel between STOCs. The called side STOC converts the document size when a called terminal is different from a calling terminal. The called side STOC asks TS to make a connection to the called party. To distinguish a facsimile call from a telephone call, a special tone is utilized to activate the facsimile terminal without ringing a bell. The coded signal is passed to FCAP, where it is decoded, modulated and sent to the terminal. The originator and the destination identification done by TS decreases undeliverable facsimile calls which may otherwise occur frequently in such a store-and-forward system as FICS. In the end-to-center communication case, a mark sheet filled with input information is transmitted to the calling side STOC in the same procedure as in the end-to-end case. STOC passes the coded picture signal to FDIC via a 64 Kb/s line. FDIC extracts marks from the coded picture signal, encodes them as a binary sequence where each bit corresponds to a mark, and passes the coded mark information to the called data center. A computer in the center interprets the meaning represented by the coded mark information. In the center-to-end case, the reverse procedure is applied. Only difference is that FDIC converts a sequence of character codes sent from the computer into the coded picture signal by using the character pattern memory.

6. Equipment structures

The LS and TS in the GSTN are equipped with new functions for FICS. Equipments in the upper network are newly designed for FICS.

6.1 Facsimile Signal Conversion and Procedure Control Equipment(FCAP)

Two types of converters, called FCAP-I and FCAP-II, are designed corresponding to MF-1 and MF-2 terminals. The FCAP specification is shown in Table 4 and the FCAP structure is illustrated in Fig.8. An FCAP is designed to accomodate 300 to 400 terminals. A 64 Kb/s bearer rate between FCAP and STOC is divided into a 48 Kb/s picture signal channel, an 8 Kb/s control signal channel and an 8 Kb/s transmission monitoring signal bits.

6.2 Facsimile Storage and Conversion Equipment (STOC)

The specification and the structure of STOC, which corresponds to the switch and the storage station in Fig.2 are shown in Table 5, and in Fig.9, respectively. A STOC has a unique interface to both types of

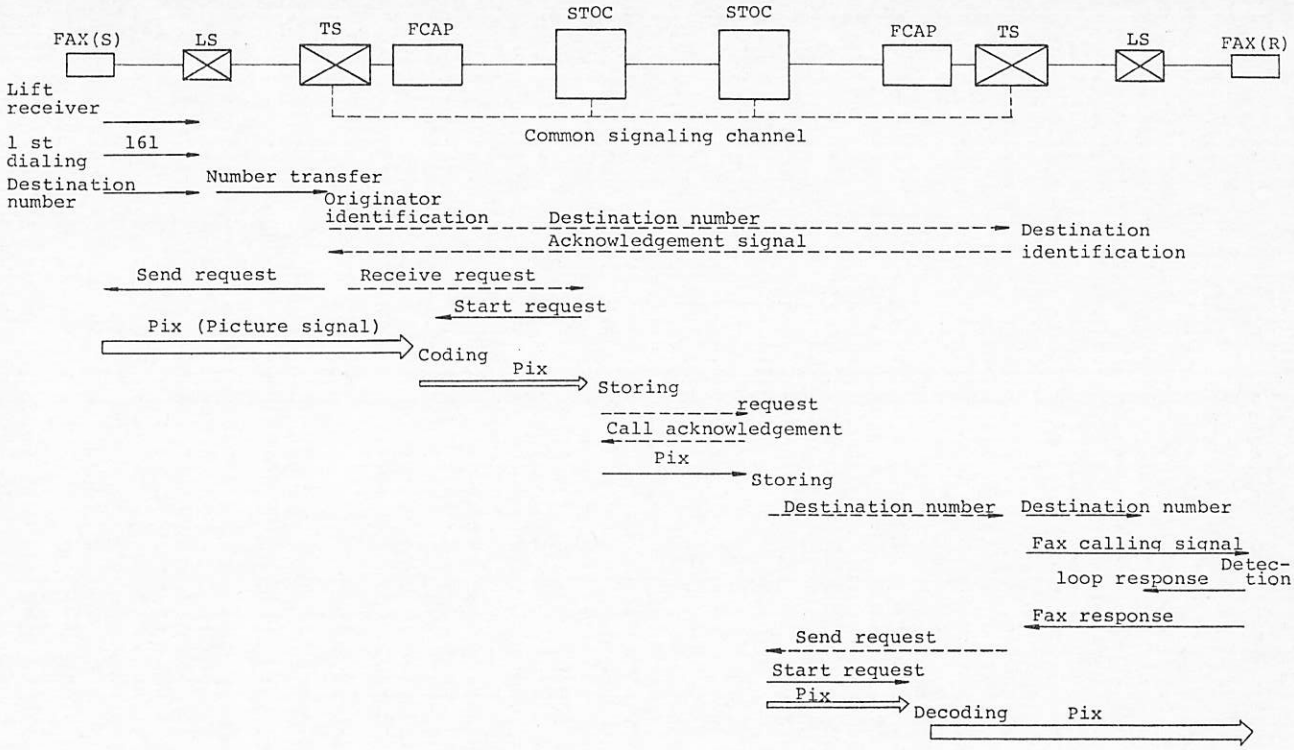


Fig.6 An example of typical end-to-end facsimile communication call procedure

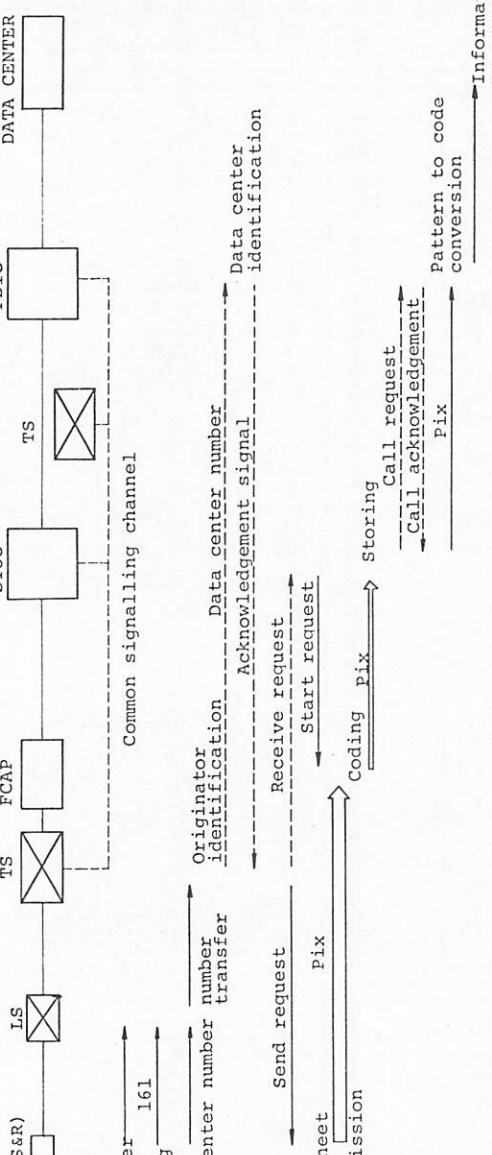
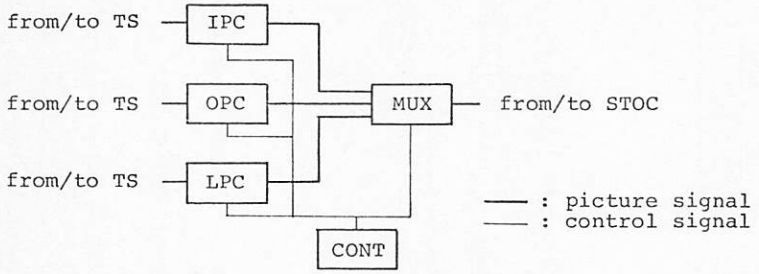


Fig.7 An example of end-to-center and center-to-end facsimile communication call procedure (in case of information retrieval)

Table 4 FCAP Specification

Items	Specifications	
	FCAP I	FCAP II
Connected terminal	MF-1	MF-2
Circuits/system	incoming circuit : 8 outgoing circuit : 8 local circuit : 4	
Interface with STOC	64 kb/s, 512 Byte/Block	
Transfer channel from/to STOC	picture signal channel : 48 kb/s control signal channel : 8 kb/s picture signal multiplexing : Packet multiplexing with frame synchronization Control signal : X-25 LAPB	
Redundancy reduction	Modified Huffman coding	fill-bit removing
Originator's number and time appending	Clapping insertion	bit pattern insertion



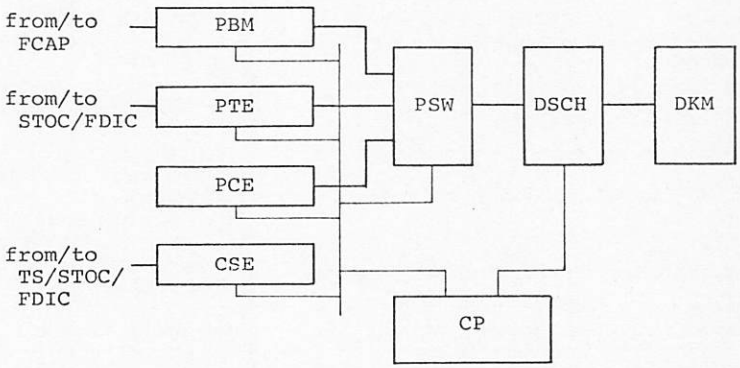
IPC : Incoming picture control circuit
OPC : Outgoing picture control circuit
LPC : Local picture control circuit
MUX : Multiplexer
CONT : Controller

Fig.8 FCAP structure

Table 5 STOC Specification

Items	Specifications
Throughput	about 57,000 BHP
Connected FCAP	80 FCAPs
Common signal channel	48 kb/s, 1+1 channels Full duplex No.7
Picture signal transfer channel	source rate : 56 kb/s, 75 channels Full duplex X-25 LAPB
Central processor execution time	average 0.6 μsec
Picture file memory capacity	400 MB/spindle × 16 spindles

* BHP : busy hour page



PBM : Picture signal buffer memory
PTE : Picture signal transfer equipment
PCE : Picture signal conversion equipment
CSE : Common channel signalling equipment
PSW : Picture signal switch
DSCH : Disk subchannel for picture file memory
DKM : Picture signal file disk memory
CP : Central processor

Fig.9 STOC structure

FCAPs and can accommodate about 80 FCAPs regardless of their types. A 400 MB/spindle packaged air-tight, tiny magnetic disk memory having 1.1 MB/s read/write speed is used as a picture file memory. Different from usual computer architecture, coded picture signals are directly transferred in 4 KB blocks between PBM and DKM via PSW and DSCH. However, the disk memory read/write area is managed by CP.

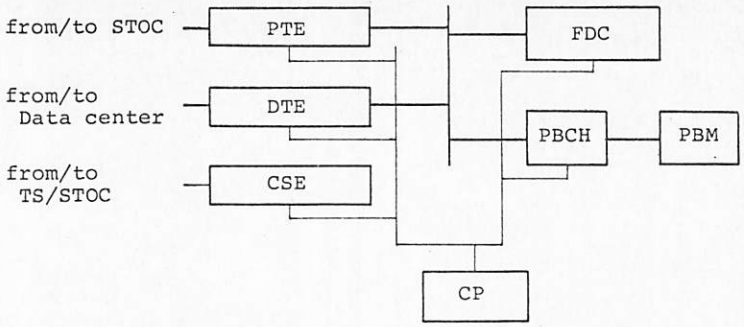
6.3 Facsimile Data Conversion and Interface Control Equipment (FDIC) The specification and the structure of FDIC designed as the media converter are shown in Table. 6 and in Fig. 10, respectively. The coded picture signal coming from STOC and character codes coming from data centers are temporarily stored in PBM. After storing a full page of document, the signals are transferred to FDC, where the picture signal is converted into mark bit sequences and character code sequences are converted into picture images. The signal thus converted is transferred back to PBM and then transmitted to data centers or STOCs via DTE or PTE. It seems straightforward that FDIC can be used as a connecting node between FICS and text communication networks such as DDX packet network.[4]

Table 6 FDIC Specification

Items	Specifications
Throughput	6,000 BHP
Common signal channel	4.8 kb/s 1+1 channels Full duplex No.7
Picture signal transfer channel	Source rate : 56 kb/s 14 channels Full duplex X-25 LAPB
Data transfer channel to data center	64 kb/s and 12.8 kb/s DCNA protocol[3]
Central processor execution time	Average 1.5 μsec
Facsimile-data conversion equipment	C/P*1 : 8 P/C*2 : 8
Picture file memory capacity	4 MB × 3

*1 C/P : Code-to-pattern conversion unit.

*2 P/C : Pattern-to-code conversion unit.



DTE : Data transfer equipment
FDC : Facsimile Data conversion equipment
PBCH : Bubble memory channel for picture file memory
PBM : Picture bubble memory

Fig.10 FDIC structure

7. Conclusion

The outline of the new advanced facsimile intelligent communication system (FICS-2) has been presented. This system is now being developed to expand the public facsimile communication services throughout the country. The followings are the summarized conclusion regarding the network architecture.
(1) The lower portion should be shared with the general switched telephone network because of user economy and investment economy.
(2) The message switching system is superior from the cost/performance point of the network to the circuit switching or packet switching system.
(3) The upper network in FICS-2 consists of FCAP, STOC and, FDIC.
(4) FCAP takes charge of terminal interfaces.
(5) FDIC takes charge of interfaces to data centers and text communication services.

Acknowledgement

The authors wish to thank their colleagues for their contributions during the research program and Dr. T.Ohmori, Mr. Ohkawara and Dr. Kanaya for the support and helpful discussions.

References

(1) Hiroyama,M., et.al., On Facsimile Application for Data Communication Network, Proc. of the fourth ICCC(1978)649-654
(2) Taniguchi,M., et.al., Experiments on Facsimile Network Service Laid over the Public Telephone Network, Proc. ICC-79(1979) 38.7.1-38.7.5
(3) Naemura,K., et.al., Data Communication Network Architecture -Objectives and Fundamental Concept, Review of ECL, Vol.27, Nos.5-6(1979)297-311
(4) Matsumoto,M., et.al., DDX Packet Switched Network and its Technology, Proc. of the fourth ICCC(1978)583-588

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

THE INTEGRATED SERVICES DIGITAL NETWORK (ISDN) AND ITS USE FOR TEXT AND DATA COMMUNICATION

Peter Bocker
Peter R. Gerke

Siemens AG Communications Group
Central Laboratories
Munich
W-Germany

The capacity of the 64 kbit/s speech channel currently being introduced in public telephone networks makes its use for text, video and data services as well as for combined voice/data communication a particularly attractive proposition. However, the full benefit of the resultant new services will only be felt if a fully digital telephone network is provided from subscriber to subscriber.

In the ISDN, the transport and standard services will be available to the text and data terminal users in a form as tailored to their needs as they are in independent networks for text and data communication, while special facilities which go beyond those provided in the ISDN for voice transmission can be introduced with the aid of operational modules. Value Added Services can be implemented in a similar manner.

1. Characteristics of the ISDN

All over the world the telephone network is making increasing use of digital methods since this results in a very economical communication network. In accordance with CCITT Recommendation G.711, pulse code modulation (PCM) techniques are being employed to convert analog telephone channels, with a bandwidth of approximately 3400 Hz, into digital channels with a bit rate of 64 kbit/s. These standard 64 kbit/s channels are used throughout digital telephone networks. There are two typical versions (Fig. 1)

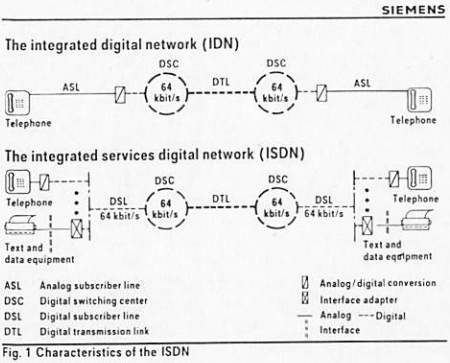


Fig. 1 Characteristics of the ISDN

a) The digital telephone network itself employs digital transmission links and digital exchanges for 64 kbit/s channels but analog methods are still used to connect subscribers to the digital exchanges, via analog subscriber terminals. The CCITT refers to such networks as Integrated Digital Networks (IDN) because the digital exchanges are connected directly to the digital transmission links, i.e. there are no A/D converters between the two.

b) Here the subscriber loops also operate on a digital basis. Thus the 64 kbit/s channels extend right through to the subscribers who can then have their telephones connected via analog/digital converters or even have other types of terminals, such as text processors and data terminals, linked up. Interface adapters are required instead of modems for connections of this sort (Fig. 1b). Since such networks can also be used by other services they are known as Integrated Services Digital Networks (ISDN).

Today, efforts are being made, both by the various European administrations and within the CCITT, to standardize the ISDN whenever necessary. As seen from the present state of the art the ISDN will have the following basic characteristics / 1 /:

1. The standard channels used throughout the network operate at a useable bit rate of 64 kbit/s (in the USA only 56 kbit/s without any restrictions). In special cases several, e.g. three, 64 kbit/s channels can be interconnected for a single call.

2. Since the ISDN is a fully electronic network the call-setup and cleardown times are considerably shorter than those in today's analog telephone network. Less than 1 s is required, the exact amount depending on the distance covered.

3. In addition to the 64 kbit/s channel a bit rate of e.g. 8 kbit/s is provided for signaling between the subscribers and the exchange. This allows considerable extension of the signaling facilities as compared with the analog telephone network. For example, it is possible for the network to inform the subscriber of the calling party's directory number or other features even while a call is in progress.

The properties mean that the ISDN is far more efficient than the present-day analog telephone network it will eventually replace. In the long run the ISDN will have the added advantage that it will be an extensive network.

The characteristics listed above refer to the basic version of the ISDN. They allow major improvements as regards speech communication: uniform, constant loudness throughout the network, improved hands-free speaking, widening of the voice band if so desired and visual display of messages from the network (e.g. calling party's directory number). As far as text and data communication are concerned, the high bit rate (64 kbit/s or a low multiple thereof) and the relatively short call-setup and cleardown times are of primary interest. It must not be forgotten that this basic version of the ISDN is a circuit-switched network and that therefore such new services as combined telecommunication can be introduced. The short transmission times make it possible to exchange texts, data or graphical material at high speeds, in order to support the spoken word, with hardly any adverse effect on the voice communication (Fig. 2).

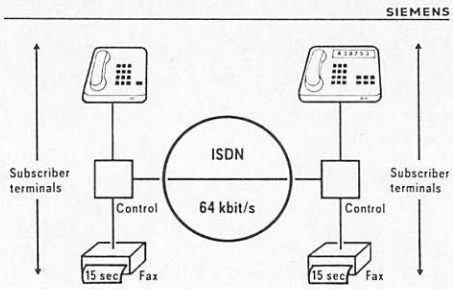


Fig. 2 Example of Combined Voice, Data and Fax Communication

This basic version of the ISDN can be extended by providing further features for text and data traffic. This is achieved by installing modules known as operational modules in all, or perhaps just in a few suitable, ISDN switching centers (Fig. 3).

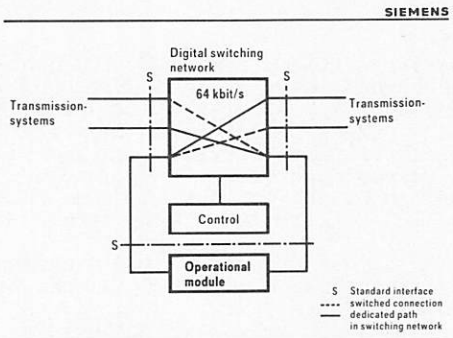


Fig. 3 Implementation of Added Values by Means of Operational Modules

The modules are connected to the 64 kbit/s digital switching networks by means of standard interfaces so that it is possible to set up connections to, from and via them. They carry out such functions as storing messages and code conversion and thus provide the network with "added values", a topic which will be discussed in detail in section 3.

2. The meaning of the word "services" in "Integrated Services Digital Network"

Because it is so closely related to the IDN and, therefore, to the public telephone network the ISDN will be a public network to which everyone will have access. As such it will offer two categories of service, i.e. transportation services and standard services.

a) Transportation services

Transportation services provide 64 kbit/s channels for carrying messages from subscriber to subscriber. The subscriber is free to utilize the 64 kbit/s channels in any way he wishes, although he must comply with the call-setup and cleardown protocols laid down for that network (Fig. 4a).

Apart from the question of establishing the tariff rates, this does not present the administrations and carriers with any special problems. However, the subscribers are faced with a more difficult situation. They and the partners they wish to communicate with must agree on the conventions to be applied when using the 64 kbit/s channels; the agreement must cover even the highest protocol levels. Since there are no prescribed standards, a large number of different, incompatible protocols will be used. This means that closed private "user islands" will develop within the public network. Intercommunication between these islands will not be possible unless the users are provided with the appropriate protocol adaption and transition circuitry. Many of today's circuit-switched networks and packet-switched networks are examples of this.

b) Standard services

Standard services are services for which the protocols have been standardized even beyond layers 1 to 3 (Fig. 4b). The public network thus opens up the possibility of communication between all who subscribe to the same services. The telephone, telex, telefax, teletex and viewdata services are the most familiar examples of such standard services. They are, or soon will be, so widespread throughout the world that there is no need to go into further details on their importance here.

The ISDN will offer a much broader range of communication facilities than today's public networks, as can be seen from the following two examples / 2 /:

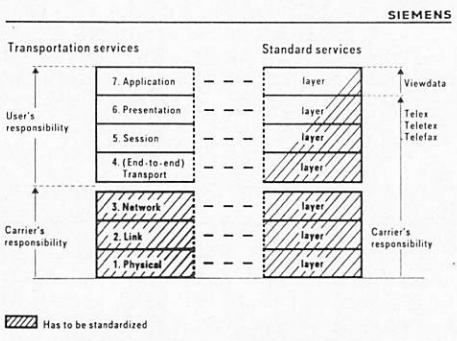


Fig. 4 Reference Model of Open Systems Interconnection According to ISO TC 97/SC 16 N 227 (Aug. 79)

1. Combined communication

The example of the new combined communication service shown in Fig. 2 depicts the transmission of a facsimile to support the voice communication. Fig. 5 shows various examples of the methods which can be employed for facsimile transmission:

a) Two 64 kbit/s channels are used for the call from the beginning to increase the voice-transmission quality. The facsimile is then transmitted on one of the 64 kbit/s channels and the voice-transmission quality is lowered to the 1 x 64 kbit/s standard while the fax is being transmitted / 3 /.

b) Only one 64 kbit/s channel is available for the call so the facsimiles are therefore transmitted during breaks in the conversation.

c) Only 56 kbit/s are used for voice transmission. There is no appreciable loss of voice-transmission quality and the remaining 8 kbit/s are permanently available for fax transmission.

d) Speech and facsimiles are transmitted alternately at a bit rate of 56 kbit/s. The remaining 8 kbit/s are free for controlling the changeover from speech to fax and vice versa.

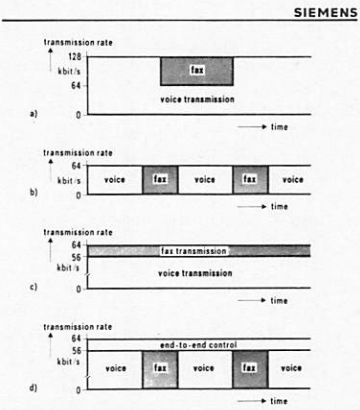


Fig. 5 Transmission possibilities for mixed voice / fax telecommunication

2. Line configurations

Fig. 6 gives a general survey of the possible line configurations in the ISDN network. The various lines differ according to the number and type of terminals (TE) connected, the requisite bit rate for each terminal (one or more 64 kbit/s channels) and, lastly, the number (n) of 64 kbit/s channels available for each line.

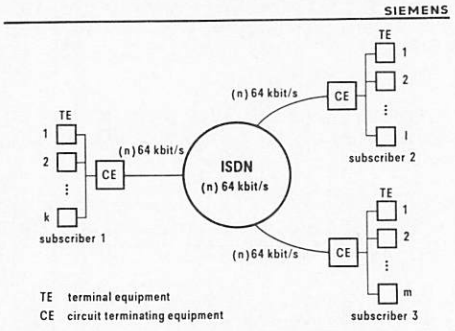


Fig. 6 Basic configuration for combined telecommunication

These examples display a wide range of communication facilities but this could be multipled several times over if every possible facility were recorded systematically. The number of user groups which are incompatible with the other groups because they utilize different facilities could become correspondingly large. This is one of the dangers of the flexibility and universality of the ISDN and clashes with the vision of a large-scale world-wide telecommunications network in the future. Such a network can, and must, allow communication between any one subscriber and any other subscriber in the world, just as today's international telephone and telex networks do.

This danger can be conteracted by selecting a limited number of attractive facilities to serve as standard services. This involves standardization right through to the top layers of the reference modul (Fig. 5b) in order to achieve compatibility through standardization. In Europe work has already begun on the selection of suitable standard services.

Beyond this ample range of standard services there are still the transportation services to meet the special requirements of individual groups of subscribers. In this way the subscribers are free to use the 64 kbit/s channels of the ISDN in a highly individual manner if they are prepared to sacrifice general compatibility and the possibility of communicating with any other subscriber on the world-wide network. If, on the other hand, subscribers to such transportation services wish to retain this facility they must provide the corresponding protocol adaption and transition circuitry themselves (Fig. 7a).

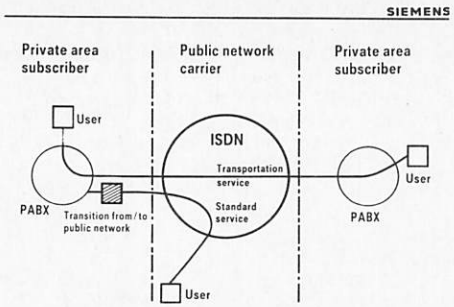


Fig. 7a Achievement of Compatibility Between Private and Public Network (1)

Compatibility can also be attained by means of compatibility services in the public network. This method is already in use for establishing compatibility between different standard services. However, it is much more difficult to make sure that the various applications of the transportation services in the public network are also compatible. In the case of the ISDN in particular, where the inclusion of voice communication results in the broad range of communications facilities already discussed, it is probably no longer possible to implement an all-embracing network compatibility service (Fig. 7b).

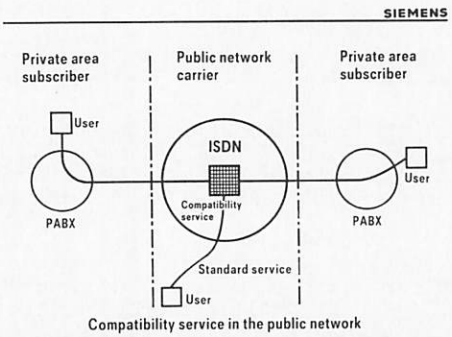


Fig. 7b Achievement of Compatibility Between Private and Public Network (2)

3. Special facilities for text and data communication

In the ISDN, the transport and standard services will be available to the text and data terminal users in a form as tailored to their needs as they are in independent networks for text and data communication, while special facilities which go beyond those provided in the ISDN for voice transmission can be introduced with the aid of operational modules (Fig. 3). Value Added Services can be implemented in a similar manner.

3.1 Transport services for text and data communication

In order to be able to establish the user-to-user 64-kbit/s channels and to utilize these for text and data communication purposes, the necessary interfaces, which have already been defined (CCITT recommendations X.20, X.20bis, X.21bis, X.25, V.25 / 3 /) must be provided between network (CE) and data terminal equipment (TE) with the aid of suitable adaption facilities located at the user terminal (Fig. 8). The definitions relate to layers 1 to 3 (Fig. 4). If additional functions are required to support the transport service, they are located in layer 4. This is e.g. the case for the transport part within the teletex service [4]. These adaption facilities are at the same time capable of converting the transmission speeds and operating modes of the data terminal equipment (50 - 300 bit/s, start-stop; 600 - 48000 bit/s, synchronous) to the synchronous 64-kbit/s channels of the ISDN.

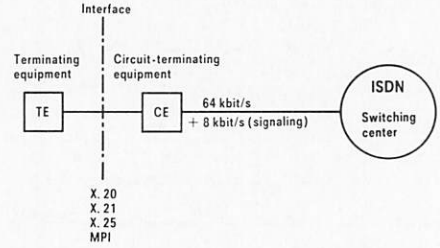


Fig. 8 Adapter Equipment on Subscriber Side

In this context the procedures for connection setup and clearing across the DTE/DCE interface are to be converted to the user signaling employed in the ISDN. If required, a switchover from voice to text and data transmission must be carried out.

The 64-kbit/s speed is higher than the maximum of 48 kbit/s offered in independent data networks according to CCITT recommendation X.1 / 5 /. Should it prove attractive for data transmission and perhaps also for facsimile transmission, it might be introduced as a new user class of service.

Over and above this, work is being done on the definition of a new interface / 6 /.

3.2 Standard services for text and data transmission

Some standard services are characteristic of text transmission in particular (see Section 2), and all users of one of these standard services can communicate with each other (telex, teletex, telefax, interactive videotex, ...). A prerequisite for this is the standardization of the communication protocol beyond layers 1 to 4.

If man is involved in these services in the role of a "text editor" (e.g. telex, teletex), only layers 1 to 6 have

essentially to be defined. Interactive videotex, where man communicates with the computer, also includes layer 7 (see Fig. 4b).

If the standard services which are being implemented today in independent networks for text and data transmission or in the analog telephone network are to be incorporated in the ISDN, then this must be done without altering them in any way. If necessary, transitions between dissimilar networks must be created.

In addition to the existing standard services, further services can be introduced in the ISDN which are specifically tailored to the new performance capabilities of this network. Especially for image transmission the high transmission rate of 64 kbit/s from user to user provides advantages: the transmission of gray tone and color images within a matter of minutes will become a reality, and new perspectives will open up for electronic mail.

3.3 Additional facilities - Value Added Services

Additional facilities can supplement the transport services and standard services. They enable the user to make use of the network in a more simple manner. The following services are possible:

- General user facilities

  These include internationally-agreed user facilities (CCITT recommendation X.2 / 7 /) such as direct call, closed user groups, abbreviated dialing, multiaddressing etc.. To these must be added national user facilities such as notification of call data, automatic retry, call diversion.

- Protocol adaption

  This allows the users to access the network not just with one procedure but with more since the network understands various user procedures.

- Message storage services

  These permit transmission under specific message control irrespective of the occupancy state of the network (hold for delivery, priorities, journalling, message diversion).

- Processor services

  These support the user in preparing and handling the texts and data to be transmitted through validity checks, editing aids, specific user instructions.

In the existing public and private seperate networks for text and data communication, these additional facilities have already been realized to an extent that depends on the particular application. In the ISDN, these facilities can be provided with the aid of the aforementioned operational modules (Fig. 3) if the switching centers do not already include them in their basic configuration (e.g. general user facilities).

4. Network load due to various services

Once the ISDN has been integrated into the IDN it must offer a viable solution for all types of traffic, including voice communication. At the same time it must be possible to distribute the costs arising in the network amongst the subscribers in such a way that the charge are roughly in proportion to the amount of use the subscribers get from the network (fair distribution of charges).

There are various factors to be taken into account when considering utilization of the network and each of these factors gives rise to corresponding costs:

a) Access to the network

Providing each subscriber with access to the network involves considerable expenditure. This outlay provides access to a functioning communications network and is, to a certain extent, independent of how much, and in which way, the network ist used.

For al long time to come ordinary telephone subscribers will form by far the largest subscriber group in an ISDN incorporating the public telephone service. The basic network access must therefore provide an economical solution for bulk telephone traffic while at the same time being suitable for other services. In return for access to the network the administrations and carriers often levy a flat-rate basic charge regardless of the number of calls made.

b) Traffic intensity

The internal dimensions of the network, i.e. the number of transmission paths and the size of the exchanges, largely depend on the amount of traffic generated by the individual services. The traffic intensity (given in erlangs, erl) specifies the number of transmission paths occupied in the busy hour and the length of time for which they are held:

$$y = \frac{c \cdot t_m}{60} \quad \text{(erl)}$$

where c is the number of seizures and t_m is the mean holding time in minutes.

There is certainly some justification for charging the subscriber a fee reflecting certain aspects of the load he places on the network, i.e. type of network components employed and holding times thereof, each time he makes a call. Many administrations and carriers adopt this approach today. The method is perfectly adequate as long as the call-setup and cleardown times are small in comparison with the mean holding time t_m. This is the case for telephone traffic, for example, where the mean holding time is 100 s.

However, the holding times for text and data traffic cover a very broad range. File transfer, for example, requires long holding times while the holding times for dialogs between the subscribers and the computer are very short. This is particularly true in a 64 kbit/s network such as the ISDN where, if the full transmission capacity is used, it takes only 0,15 s to transfer the 1200 or so characters contained on an average VDU screen. In this case the load placed on the network is

minimal in terms of the traffic intensity although the equipment required for setting up and clearing down the connection has to operate in exactly the same way as it would for a telephone call lasting much longer.

The following figures will help clarify this relationship: Every telephone call (with a mean holding time t_m of 100 s) places a mean load of $2{,}8 \cdot 10^{-2}$ erl on the network (1 s is adequate to $2{,}8 \cdot 10^{-4}$ erl). A data connection for transactions in the interactive mode (question and answer), with a mean holding time (t_m) of 2 s, places a load of $5{,}6 \cdot 10^{-4}$ erl on the network (traditional circuit switching, TCS). If, on the other hand, a separate connection is set up and cleared down for both question and answer the load on the network is reduced to approximately $8 \cdot 10^{-5}$ erl per transaction (transmission time of the messages about 300 ms), even though the call-setup and cleardown load is doubled (fast circuit switching, FCS). (Any partial seizures occurring while the connection is being set up are ignored.)

c) Number of calls

It is clear from the example that in the ISDN greater significance must be attached to the number of calls made than to the traffic intensity. It is possible that approximately the same proportion of calls will be made for both the telephone and text/data services offered by the ISDN even though telephone services will outweigh all other forms of traffic in the future too as regards the number of subscribers and the load places on the network. In other word, ultimately the major influence on the size of the ISDN and the load carried by the network will be the telephone traffic, although data traffic will also affect the dimensioning of the control elements required for setting up and clearing down the connections. It is therefore only fair that the tariff structure should demand that a larger share of the costs for the control elements in the ISDN be borne by traffic with a very short holding time t_m. In order to do this the network must be informed what type of call is being made each time a connection is established, including those for transportation services.

It should also be noted that if services above and beyond those including in the basic version of the ISDN are used, i.e. by means of operational modules, supplementary charges can be made without difficulty.

5. Introduction of the ISDN and chronological development

Naturally, it will be a long time before the IDN and ISDN have been introduced in their entirety throughout the world. However, it is possible to install a relatively "thin" fully-digital overlay network at the same time as going about the more lengthy task of converting the public telephone network to digital methods. Subscribers who wish to enjoy the benefits of the ISDN as early as possible can then be connected to the overlay network. The majority of these subscribers will come from the business sector. Fig. 9 indicates how the IDN and the ISDN, in its overlay network form, can develop side by side, sharing the available digital exchanges and transmission paths, before ultimately combining to form a single uniform network.

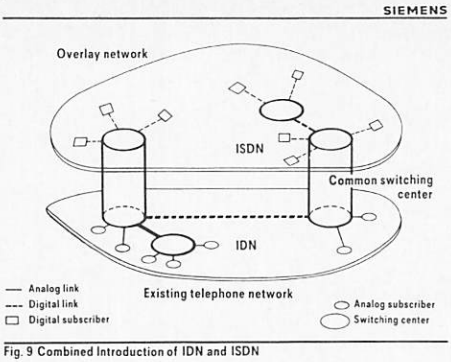


Fig. 9 Combined Introduction of IDN and ISDN

However, the first major step to be taken towards on ISDN is for the CCITT to recommend the international standards discussed earlier. This involves specifying standard services on the one hand and recommending suitable technical parameters and the requisite protocols for the utilization of the 64 kbit/s channel on the other hand. Considerable progress can be expected in this direction by the end of the coming CCITT study period (1984).

References

/ 1 / Gerke P.; Bocker, P.: DTN, an all digital telephony network for voice, data, text and fax communication. International Switching Symposium, Paris, 1979, pp. 773 - 780.

/ 2 / Gerke, P.: The integrated services digital network (ISDN) as a basis for combined telecommunication. Eurocon, Stuttgart 1980, pp. 233 - 236.

/ 3 / CCITT: Recommendation X.20: Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for start-stop transmission services on public data networks. Orange Book, Vol. VIII.2, Geneva: ITU 1977.

CCITT: Recommendation X.20bis: V.21-compatible interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for start-stop transmission services on public data networks. Orange Book, Vol. VIII.2, Geneva: ITU 1977.

CCITT: Recommendation X.21: General purpose interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for synchronous operating on public data networks. Orange Book, Vol. VIII.2, Geneva: ITU 1977.

CCITT: Recommendation X.21bis: Use on public data networks of data terminal equipments (DTEs) which are designed for interfacing synchronous Series V recommendation modems. Orange Book, Vol. VIII.2, Geneva: ITU 1977.

CCITT: Recommendation X.25: Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode on public data networks. Orange Book, Vol. VIII.2, Geneva: ITU 1977.

CCITT: Recommendation V.25: Automatic calling and/or answering equipment on the general switched telephone network, including disabling of echo suppressors on manually-established calls.
Orange Book, Vol. VIII.1, Geneva: ITU 1977.

/ 4 / CCITT: Recommendation S.70: Network independent basic transport service for Teletex. Yellow Book, Geneva: ITU (to be published 1981).

/ 5 / CCITT: Recommendation X.1: International user classes of service in public data networks. Orange Book, Vol. VIII.2, Geneva: ITU 1977.

/ 6 / CEPT, Special Group ISDN (GSI) Report of meeting No. 4 of Special Group ISDN (GSI) in Lausanne, 20-24 October 1980. Doc. T/CCH(80)13, Doc. T/GSI(80)92.

/ 7 / CCITT: Recommendation X.2: International user facilities in public data networks. Orange Book, Vol. VIII.2, Geneva: ITU 1977.

USER REQUIREMENTS FOR COMPUTER MESSAGING

Chairman - E. Feinler

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

USER EXPERIENCE AND EVOLVING DESIGN IN A LOCAL ELECTRONIC MAIL SYSTEM

J. Bruder, M. Moy, and A. Mueller
Hewlett-Packard Company
1501 Page Mill Road
Palo Alto, California 94304

R. Danielson
Department of Electrical Engineering
and Computer Science
University of Santa Clara
Santa Clara, California 95053

A local computer mail system, if use is not forced by corporate management, runs the risk of lying idle unless it is well-matched to its operational environment. This paper describes such a system, which was developed for internal use at Hewlett-Packard Company, in response to a specific request from a small organization within the company. However, the system was designed, and has proven, to be applicable in a broader context. Implementation has occurred in a series of gradual steps, which allowed user experience to direct successive versions, and required minimal commitment of resources at any stage. Subjective user comments on features of two versions, as well as objective statistics on use of the current version, are presented and discussed.

INTRODUCTION

The potential benefits of electronic mail systems have been described by a number of authors (e.g., [1], [2], [3]). The positive contributions usually attributed to mail systems include:

- permanent records of communication,
- asynchronous communication,
- freedom from interruption,
- release from geographic restrictions, and
- optimal use of sender's and receiver's time.

Most of these claims have been made in connection with mail systems serving widely separated communities. It is not immediately apparent that these benefits must accrue in a local electronic mail system.

Without the incentive provided by geographical dispersion, other factors become proportionately more important. Such factors include:

- the characteristics of the user community,
- the nature of the messages, and
- accessibility of the mail system to users.

The need to modify principles used successfully in other systems, based on these characteristics of the local environment, puts a certain element of risk into the implementation of a local message system.

The Hewlett-Packard Company (HP) has considerable experience with geographically dispersed electronic message systems, having implemented in 1971 an international

computer communication system called COMSYS [4]. This system uses voice grade telephone lines for batch transmission of messages at scheduled times between COMSYS Central in Palo Alto, California and HP plants and offices around the world.

Message traffic currently averages 140 million characters daily, of which some 15% are "administrative messages" of the type usually associated with electronic mail. COMSYS, however, uses traditional office mail facilities for distribution of messages at the destination site.

The local electronic mail project described in this paper began with a request to the Office Utilities Group (OUG) of HP's Corporate Information Systems Department from HP Labs for a simple message system to be used within the Labs for project management.

PRE-IMPLEMENTATION CONSIDERATIONS

OUG, as the name suggests, is chartered to develop various programs associated with office activities (e.g., editors and text formatters) strictly for internal use by different HP divisions. Since the various divisions are relatively autonomous, OUG is in a position similar to an external vendor. For its products to be successful, they must be economically usable by a wide range of potential users with different characteristics.

Despite the proven effectiveness of COMSYS, the concerns expressed in the preceding section regarding local computer-mediated mail systems dictated a cautious approach to satisfying the request. It was decided to use a phased approach to implementation, in which steps between phases could be taken with minimal resource investment, based on user experience with the previous version.

Although the prototype version of the mail system was intended for a limited, relatively sophisticated audience, it was hopefully to be the forerunner of a system to be made available to the entire company. Thus the user community was dictated, from the start, to consist of all levels of personnel (clerical/secretarial, programmer or other professional, and management) with all levels of computer sophistication. This meant the system had to be simple enough to be quickly learned by naive users, yet not be cumbersome for experienced or sophisticated users.

A second dictate was that the system would use only existing hardware. Essentially, this implied execution on HP3000 computers equipped with HP 264x terminals, although it was decided to support any terminal attached to an HP3000.

Further, it was decided to maintain messages in a separate message database, organized by receiver name, rather than simply storing them as text files in a receiver's own filing space. There are two reasons for this approach. One is that many potential users have more than one signon. Centralizing the database would allow them to access all their mail from any account. The second is the opposite phenomenon: sometimes several different users sign on to the same account. The centralized approach provides these users privacy.

NORMAN I

This first version of NORMAN (Network Oriented Routing of Mail with Automatic Notification) was developed in a very short period of time and installed at HP labs in November, 1979. The implementation team consisted of first one, and later two, programmer(s) with no previous experience with computer mail systems.

Despite the acronym, NORMAN I was limited to users on a single HP3000. Its major characteristics were:

- A simple, command-driven user interface. Nine commands were available (for message management, as well as HELP access and receiver directory services), but each had only limited scope and flexibility.
- Messages could be up to 20 lines long, and could be sent to up to 20 receivers. However, each receiver would have her/his own copy of the message stored in the database. Messages were stored on a line-by-line basis.
- Only rudimentary editing was available. Individual lines could be entered and modified until RETURN was pressed, which stored the line in the database. Once there, lines could not be changed or deleted (without deleting the whole message).
- Automatic notification of receipt of mail. This would be to the logical device number of the user's terminal, or to whichever terminal s/he was using, if NORMAN were run while logged on to a particular session.
- Automatic purge of messages more than one week old, unless they had been explicitly SAVEd by the user.
- Communication only with individuals explicitly entered in NORMAN's receiver database.
- Optional passwords, associated with the individual user's name in the receiver database.

The user population for this initial installation was about 40 people, with a user/terminal ratio of about 4/1. NORMAN received only limited use over a five month period. Major complaints from users of the system were:

- Message length and number of receiver restrictions were too severe.
- A distribution list facility was needed to eliminate typing of receiver names.
- More extensive message editing capabilities should be offered.
- The mail system should be accessible from other machines.

NORMAN II

Based on these user comments, and further investigation of features of other mail systems, a second version of NORMAN was implemented. This version represented a complete reorganization of the database used with NORMAN I, as well as a complete redesign of the code to improve modularity and allow for easier expansion of capability. It is the current mail system for most of the user community.

NORMAN II addresses all the user complaints from the previous system, with the exception of editing, and includes a few minor enhancements. Major features of the new version are:

- Unlimited message length and number of receivers. Message text is compacted to save space in the database. Only one copy of the message is retained regardless of the number of receivers. This copy is deleted only when all receivers and the sender have PURGEd the message.
- A distribution list utility allows users to create and maintain both public (usable by anyone) and private (usable only by the creator) lists of recipients.
- Access to NORMAN from any computer in a Distributed System/3000 network. There is still only a single NORMAN program, with its

corresponding database, but it may be accessed from remote machines. Notification was also extended to parts of the DS/3000 network.

- A "public signon" allows users to access NORMAN without having knowledge of the operation of the network control protocols.
- Options were added to most commands to increase their flexibility and power. Additional commands were added for the distribution list facility and to allow formatting and printing of messages to system lineprinters, or to local printers slaved to a terminal.
- Instrumentation of the program to maintain statistics on messages, sessions, command usage, etc. (See below).

NORMAN II was initially installed in OUG itself in March, 1980, and has gradually been extended to the rest of Corporate Information Systems. HP Labs has their own NORMAN system using a computer and database separate from the CIS network. It has received extensive use over a six month period.

Detailed discussion of use of various features of NORMAN II is presented in the next section. Subjective comments from users indicate the following areas are major candidates for enhancement:

- Editing facilities are still minimal. This is the major source of user complaints.
- Lack of multiple-level notification. While NORMAN can be run from any computer in a local network, notification occurs at most one level away from the main NORMAN machine.
- Lack of a filing and retrieval system for messages. NORMAN allows messages to be SAVEd, to prevent their deletion. SAVEd messages may be READ and PRINTed, but cannot be resent.
- Minor requests for additional commands and features, such as message annotation and forwarding, a broadcase facility to all users, and the ability to cancel a message even though it has been read by some of the recipients.

USER EXPERIENCE

The statistics cited in this section were collected from two separate user communities:

- Corporate Information Systems Department (CIS), with 120 NORMAN users, and a user/terminal ratio of essentially 1/1. The figures for CIS reflect the most recent eight weeks of use.
- HP Labs (HPL), with 85 users and a user/terminal ratio of about 3.5/1. Statistics cover the most recent seven weeks of use.

Overall Usage

Casual observation of NORMAN use indicates that local mail is used to augment, and often replace, routed hardcopy memos and verbal messages. As would be expected, use of the mail system is most intense during normal working hours.

However, some individuals make extensive use of the system on weekends to provide

easy communication with co-workers who follow more regular schedules. For this reason, data averaged "by day" in the Tables below are presented for both five and seven day work weeks. Statistics were collected only as weekly totals, with no actual breakdown on a day-by-day basis.

The notification feature is an important aspect of use of the system. It seems to spark user curiousity and decrease time lapse between message sending and reading. Systems use decreased noticably during a short period when the notification feature was not working.

Table I presents information on number of messages, number of sessions, and number of recipients for both environments. System use is relatively high, significantly more so in the CIS environment where terminals are much more available.

TABLE I

GENERAL USE STATISTICS

	CIS	HPL
Number of messages	1344	545
per day (7)	24.0	11.1
per day (5)	33.6	15.6
Number of sessions	3470	1459
per day (7)	62.0	29.8
per day (5)	86.8	48.6
Number of recipients	4342	912
per message	3.2	1.7

This difference holds even when message and session figures are normalized by dividing by the number of users. CIS users send 1.5 times as many messages per user as HPL users, and log on to NORMAN 1.4 times as often. They also tend to send a message to nearly twice as many recipients.

This is not surprising. Where users have easy access to terminals, the computer-mediated communication environment is simply more dynamic. The notify feature is also more effective with a 1/1 user/terminal ratio.

Users still consider the message system to be effective in the environment with several users for each terminal. However, the difference in statistics outlined above creates significant concerns for environments where the ratio is very large (10/1 or greater).

Message Length

Table II presents information on message length. Note that messages are quite short: a clear majority are less than 10 lines, and virtually all are 23 lines or less, which is one display screen on the HP 264x terminals commonly used.

The number of bytes per message is significantly less than reported elsewhere (e.g., in Texas Instrument's international message network [5]). This may be due to the local nature of the communication, in that lengthy messages are delivered face to face and orally, rather than typing them into the message system. Alternatively, it may be the result of the poor editing facilities and inability to send from previously composed files.

Messages are nearly twice as long in the terminal-rich environment. This may be due to pressure, in the HPL environment, to rapidly relinquish a terminal if other

users are waiting.

TABLE II

MESSAGE LENGTH DISTRIBUTION

	CIS	HPL
Total number of lines	9911	2158
avg per message	7.4	4.0
Total number of bytes	413871	88948
avg per message	307.9	163.2
Messages with length (in lines)		
< 10	1070	498
(%)	(79.6)	(91.4)
10 - 23	229	39
(%)	(17.0)	(7.2)
24 - 46	40	7
(%)	(2.9)	(1.3)
> 46	6	1
(%)	(0.4)	(0.2)

Number of Recipients

The distribution of recipients per message is summarized in Table III. Table I indicated the average number of recipients per message is low. The distribution figures show that most messages are sent to a single individual. This is unexpected, considering the demand by NORMAN I users for a distribution list capability.

TABLE III

MESSAGE RECIPIENT DISTRIBUTION

	CIS	HPL
Messages with recipients		
1	1030	491
(%)	(76.6)	(90.1)
2 - 10	222	45
(%)	(16.5)	(8.3)
11 - 25	72	10
(%)	(5.4)	(1.8)
> 25	21	2
(%)	(1.6)	(0.4)

These two factors indicate users feel that constantly typing in even a small number of addressees is a bothersome task.

It should be pointed out that figures on recipients may reflect the typical HP work structure of relatively small project groups with close intra-group interaction.

Session Length

Session length distribution figures are presented in Table IV. Again, most sessions are very short, less than six minutes, with a tendency for slightly longer sessions in the terminal-rich environment.

TABLE IV

SESSION LENGTH DISTRIBUTION

	CIS	HPL
Session Length (in minutes)		
< 6	2489	1256
(%)	(71.7)	(85.2)
6 - 15	632	90
(%)	(18.2)	(6.2)
> 15	389	2
(%)	(11.2)	(8.3)

Command Use

Table V summarizes the use of the different commands provided by NORMAN II. Note that four commands account for over 80% of the use in both the CIS and HPL environments. These are the basic commands used for reading, sending, deleting, and displaying summaries of new and previously stored messages.

TABLE V

COMMAND USE DISTRIBUTION

COMMAND	PERCENT USE	
	CIS	HPL
CANCEL	0	0
DIRECTory	2	2
DISTRIBution list	1	0
HELP	5	3
NAME	2	9
PRINT	0	0
PURGE	31	31
READ	25	22
REPLY	1	0
SAVE	1	0
SHOW existing mail	17	15
SEND	10	13
SET notify data	0	0
Commands issued per week	1900	730

One interesting characteristic is that there are more PURGE commands issued than READ commands. This might be interpreted as recipients throwing away mail without reading it.

However, this behavior is seldom observed in connection with the NORMAN system. The probable explanation is that recipients tend to issue one READ command for all new mail (READ NEWIN) but PURGE single messages. Table V contains counts of commands issued, not numbers of messages affected by each type of command.

Two commands (DISTRIB and DIRECT) relate to message recipients. Note that despite the requests for its inclusion in NORMAN, there is only small use of the DISTRIBution list facility. Combined with the relatively large number of lists on the system, this would indicate that lists are established once, and then remain fairly constant.

The low use of the DIRECTory command implies that most individuals are readily aware of the system names of the people they are sending to. This is surprising, since users express high initial dissatisfaction at having to type the recipient's name exactly as it has been entered in NORMAN. Presumably they learn the names of people they communicate with often, set up distribution lists for groups of recipients to whom they send less frequently, and only look up recipients of infrequent messages.

Two surprising figures are the low use of the SAVE and REPLY features. This may be due to the fact that receivers are not offered a choice of the operations immediately after reading a message. Instead, they must specify which message the command applies to (with the message number) after reading through all new mail. People also tend to forget REPLY exists, and use SEND instead.

This holds true for PURGE, as well, without a noticeable effect on its application. It may simply be that most messages are not important enough to SAVE, which may, in turn, be due to the local nature of the communication.

With REPLY, however, receivers are more likely to remember the name of the sender of a note than the corresponding message number. It is thus easier to SEND a message back to the person by name rather than REPLYing to a number.

Finally, the relatively low use of the HELP facility is surprising, in view of the importance placed on the existence of on-line HELP in most discussions of user interfaces [6, 7]. The weekly totals do indicate a significant increase in use of HELP (five to seven percentage points) in the week immediately following installation of NORMAN in a new group. However, usage returns to average levels in the next week.

This may mean that the need for HELP can be obviated by better training of users at the time of installation. As a practical matter, however, one cannot expect to train every user of an electronic mail system. Thus the HELP feature remains important as a tutorial for users who wish to learn the system on their own, and as an update medium for existing users as new features are implemented.

FUTURE ENHANCEMENTS

The experience with NORMAN II has indicated that a strictly local electronic mail system will be used and can be an effective communication medium. This provides incentive for continued development of a local mail system.

As of this writing (mid-September, 1980), a moderately enhanced, second version of NORMAN II is operational and available to members of OUG. It responds to some of the user comments discussed in section 4, featuring:

- Immediate disposition of each message as it is read or sent (option to SAVE or PURGE a message).
- A delayed send facility, to allow users to send reminders of future events (meetings, etc.) to themselves or other recipients.

- A broadcast send facility, to send mail to all recipients in the database without a special distribution list. This latter facility is offered with some trepidation, due to concern about possible explosions of junk mail.

A third version is planned for initial installations in roughly two months. This will probably be the last version of NORMAN II.

Enhancements planned include:

- A solution to the limited editing facility. Two approaches will be used, as well as retaining the current single-line editing capability, which is sufficient for very short messages.

  The first approach is block mode entry of message text. This allows the sender to use the local editing capability of most terminals attached to HP3000s to compose a message in the terminal before transmitting it to NORMAN. The incoming message will be converted to a line-by-line representation *via* a simple routine and stored in the existing database.

  The second approach is to allow message text to be input from an existing file created using a text editor. The input file will be parsed and stored in the same manner described above.

  These enhancements should remove the major remaining user dissatisfaction with past versions of NORMAN.

- Capability to annotate and forward messages received. The forwarded version will be a new message, with the original remaining in the database.

- Capability to send a COMSYS message from NORMAN. This will not be an electronic connection between the two systems, but rather will make use of an existing OUG product (COMGRAMS [8]), which allows COMSYS messages to be created on HP3000s and batches them onto tape. The tape is then manually carried to the COMSYS machine.

Following development of version 3 of NORMAN II, design will commence on NORMAN III. This version will be a complete redesign of the mail system. The major difference will be elimination of the central network database. Instead, NORMAN III will feature local databases and separate copies of the NORMAN user interface routines on each machine in the network.

When necessary, messages will be shipped to the receiver's "notification system" using a separate TRANSPORT routine. This will be a low-level utility in its own right, masking the characteristics of the network protocols and individual network architecture from application programs and other utilities running in the distributed system environment.

It is also probable that NORMAN III will subsume the current role of COMGRAMS. End users will see but a single system for computer mediated communication anywhere within HP. The actual medium for message transmission will depend on the relation between the locations of the sender and receiver.

SUMMARY

This paper has described past and future actions in the development of a local electronic mail system. The system has been developed in a relatively short time (less than a year for a functional system accepted by the user community) with minimal investment of resources (the development team never consisted of more than

three programmers, some of whom had additional major responsibilities). The phased design allowed user experience to motivate developments at the next stage, ensuring that features added were really those wanted by the user community.

In addition, detailed statistics related to use of the current version have been presented and discussed.

ACKNOWLEDGEMENT

The authors wish to thank Luis Hurtado-Sanchez for discussions and guidance during the design and implementation of the various NORMANS.

REFERENCES

[1] Bair, J. H., Communication in the Office-of-the-Future: Where the Real Payoff May Be, in: Proceedings of the Fourth International Conference on Computer Communications (1978).

[2] Engle, G. H., Groppuso, J., Lowenstein, R. A. and Traub, W. G., An Office Communication System, IBM Systems Journal 18:3 (1979) 402-431.

[3] Uhlig, R. P., Human Factors in Computer Message Systems, Datamation 23 (1977) 120-126.

[4] Taylor, H. D., HP - Communication System, in: NTC '77 Conference Record, Vol. 3, 21:6 (1977).

[5] EDP Analyzer, 17:9 (1979).

[6] Hansen, W. J., User Engineering Principles for Interactive Systems, in: Proceedings of the 1971 AFIPS Fall Joint Computer Conference, 523-532 (1971).

[7] Martin, J., Design of Man-Computer Dialogues (Prentice-Hall, Inc., Englewood Cliffs, N. J., 1973).

[8] Office Utilities Group, Hewlett-Packard Corporate Information Systems, COMGRAMS System Overview (August, 1979).

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

COMPUTER MEDIATED MESSAGING FOR INTERACTIVE PURPOSES

Wolfgang Dzida

Institut für Software-Technologie
Gesellschaft für Mathematik und Datenverarbeitung
D-5205 St. Augustin 1
West Germany

One of the major features of communicative acts in project teams is called feedback control. It will be shown in this paper how feedback control may influence the planning activities of team members, and which feedback events can be assisted by a special purpose message system. It is argued that the design of this communication tool depends on a psychological job and interaction analysis in the user environment. Feedback events are associated with certain types of messages, and with communication rules which guide the user of a message - system in cooperating with other team members. Some requirements are formulated for furthering the design goal of responsiveness.

1. INTRODUCTION

One of the most striking benefits of a computer based message system (CBMS) is the reliable transport of information. However, message transport is but one of the purposes of a CBMS. This paper proposes some interactive purposes of message exchange which are derived from observations of team work, in particular, in a team of software engineers.

Before certain services of a CBMS may be implemented in a software engineering environment, a psychological job and interaction analysis should be carefully carried out. The results of this analysis will contribute to a specification of requirements for computer-assisted communication between team members.

Communication necessarily takes place where people have to coordinate their plans and activities. If people work together in a team, diverse datebooks, plans and tasks of each team member should be organized or integrated harmoniously. Lack of efficiency in team work may be due to a negligent management of task coordination; thus the members of a team waste time and effort in complaining about mismatched plans and results.

There are at least two different ways to solve the coordination problem. In a centralized communication structure of a project team it is the chief who has control of the distribution of tasks among the members of his group, and he is mainly responsible for coordinated task performance. If the communication structure is decentralized, the members of the group become more responsible for a balanced and task-effective cooperation themselves. In any case of project

management, interaction strategies get implemented, some of which may be called project standards, for instance for review and inspection (Bruggere (1979)). The group members have to follow some rules of documentation and communication, and the corresponding message exchange between the members must be assisted by an appropriate tool. A CBMS can be taken as an appropriate one.

In this paper some requirements are formulated. They are proposed in order to improve the quality of message exchange in a software engineering environment that is characterized by division of labour, where the major problem is that of cooperation between different specialists sharing different tasks and information.

2. COMMUNICATION AS FEEDBACK CONTROL

2.1. FEEDBACK IN SOCIAL INTERACTION

As the task structure in a project team becomes more complex, the effort involved in gathering, analyzing and preparing information may turn out to be not negligible. "Actually, quite a bit of iteration takes place between gathering and analyzing" (UHLIG et al., 1979); and it may be added that communicative acts are involved in this iterative process, if several members of a team are provided with different information sets. As a practical example, consider the incremental design of several modules by programmers in a project team: the raw design has already been finished, but the fixing of design details and the agreement among the programmers is a matter of "disciplined" communication (c.f. CASHMAN & HOLT (1980)). Each programmer has to consult with his fellow programmers insofar as the planning and construction of one module depends on the design of other modules. This communication between team members is taken to be feedback, and it is controlled by means of a CBMS.

The purpose of this paragraph is to show that feedback control, properly used as a descriptive term, can be accurately used as a concept for CBMS human interface design. Feedback in a project team is "a direct perceptual report of the result of one's behavior on other persons" (ENGLISH & ENGLISH (1958)). In a project team where communication between the members of the project is assisted by a CBMS, the "report of the result of one's behavior" is not directly perceptual but mediated by the display of the CBMS. Control on this feedback is warranted by the CBMS which takes charge of certain feedback events. A general feedback event that should happen during a planning process will now described.

2.2. A GENERAL FEEDBACK SITUATION

Communication necessarily takes place when members of a team make a plan. Fig. 1 illustrates feedback control in the environment of a user of a CBMS who makes a plan. The user defining a task (for instance a module) initiates message exchange at a CBMS between himself and someone else, who checks an aspect of practicability. Feedback is given to the person who has initiated the message exchange. Feedback involves a reply suggesting some constraints that may help to redefine a task or to define it in more detail. The more degrees of freedom there exist in planning a task, the more feedback control is necessary. Message exchange takes place in order to keep control where the planning and performing of a task are mutually dependent.

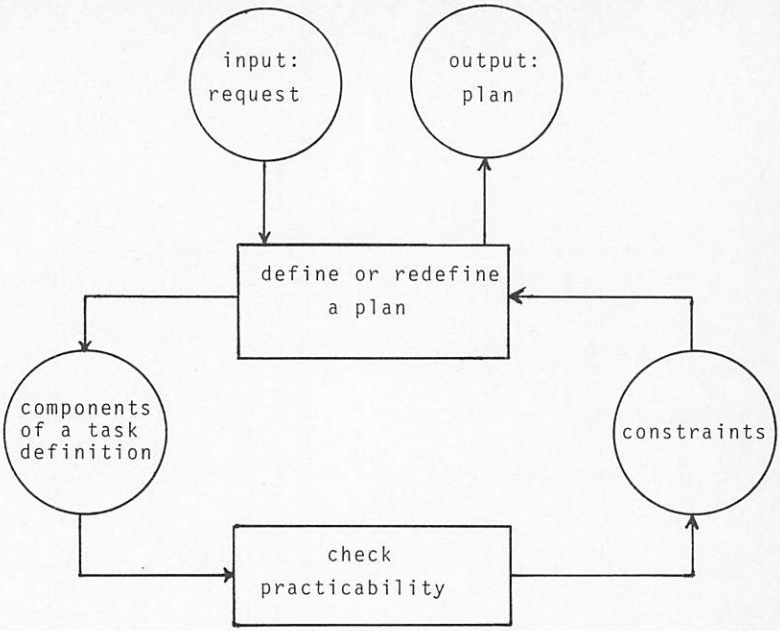


Fig. 1: Planning of a task under consideration of performance conditions; circles indicate messages and rectangles indicate information processing in a feedback control system for task definition

An empirical example has been described by TEITELMAN (1979), who reported upon a programming environment and an integrated message exchange over the ARPA network: In a user's programming environment there is a display for sending and receiving messages through a collection of display windows. TEITELMAN reported that "such electronic mail has become the preferred form of communication for questions, bug reports, suggestions, etc." (p. 173). The author illustrated a quite normal case of interaction between team members: One member asks another for help by sending him a message; the recipient of this request for help checks the described situation including a part of the sender's program and finally answers by suggesting an appropriate tool which would help.

The concept of feedback is appropriate to this case of interaction; it allows one to speak of the "purpose" of communicative acts (MILSUM (1968)), and thus of the purpose of a CBMS that is established to assist these communicative acts. The general purpose of message exchange in a project team is to move the team towards a mutual goal. Feedback corrects poorly adapted plans and the requested design comes along faster.

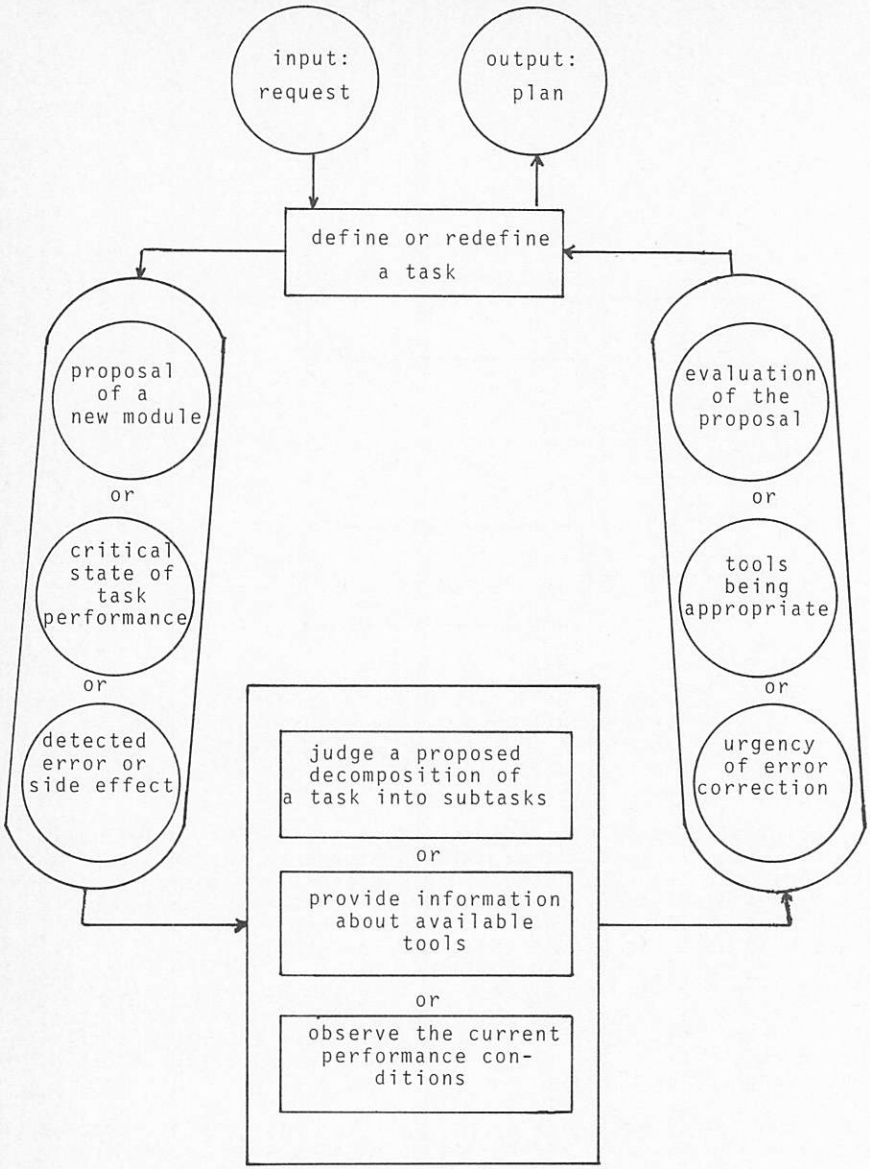


Fig. 2: Planning a task with regard to some performance conditions; circles indicate components of a plan to be communicated as well as performance constraints; rectangles indicate information processing in a feedback control system for task definition

2.3. SOME SPECIAL FEEDBACK EVENTS

The analysis of feedback events leads to the decomposition of a general feedback event, such as "check of practicability" (fig. 1), into elementary feedback events which are involved in the general one. The intention is to go beyond TEITELMAN's example of message exchange in a software engineering environment. The decomposition of the general feedback event into elementary ones leads to a classification of communicative acts in a project team, each class of which demands a distinctive message display at a CBMS.

In fig. 2 feedback events and related communicative acts are specified. The decomposition is derived from a model of man-computer interaction (DARLINGTON et al., (1981)). Three classes of human activities are there distinguished. According to these activity classes three feedback events are defined:

1. judging a proposed decomposition of a task into sub-tasks;
2. providing information about available tools;
3. observing the current performance conditions.

In order to illustrate each of these feedback events, interaction between team members as it occurs in a project team will now be described.

Case 1: A programmer is designing a module and comes to the conclusion that what he is doing is not right; he decides to redefine his task and sends a message to his colleagues. A new splitting of a module is proposed to them. These colleagues give feedback to this initiative. They send replies, each of which contains comments on the proposal. One team member may inform his colleagues that he has solved a similar problem in the past, and refers to an existing module. Duplication in effort can be avoided by this feedback event, or it can be supported a joint estimation of the utility for creating a new module.

Case 2: A programmer asks his colleagues whether one of them knows of a tool that might overcome the critical state of a problem. Feedback is given to this request, in that the programmer is informed about appropriate tools. This feedback event reduces the burden on a less experienced programmer by giving him the assistance of the more experienced.

Case 3: A programmer detects an error in a module that has the state of being "integrated". All of his colleagues are going to be immediately informed. Feedback is given to the programmer by giving him evaluations about the urgency of error correction. This feedback may be valuable, if it helps to avoid a standstill in the whole group of programmers.

2.4. TYPES OF MESSAGES

The list of feedback events is by no means complete. Investigations into the interaction of a project team will reveal other kinds of communication issues which could be integrated into the proposed classification. Each team member should get acquainted with this classification so as to improve the opportunity for a team member to send or to receive the information necessary for activities to be coordinated in a production team.

One may argue that these feedback events have nothing to do with the human interface design of a CBMS. This paper is intended to show that an analysis of feedback events may be a prerequisite for the interface design, if one is interested in going beyond a general-purpose interface for the composition and the exchange of messages.

In connection with the above presented three classes of feedback, one may distinguish three types of interfaces for the sender and the recipient respectively. Thus, each message is not only defined by a message header and a message body but by an additional striking feature: the message type. A message type indicates the pragmatic status of a content to be communicated. Such a status might be a proposal, a request for help, the announcement of a disturbance, a comment, an advice, etc. (fig. 2). The message types serve different interactive purposes. Each message type is strictly related to well defined communication rules, created by an agreement between the team members. Some of these rules will be discussed in section 3.2.

The user of a CBMS learns to classify his messages according to the different feedback events in a production team. Messages that are not classified are exchanged by using the general-purpose message display.

The classification of messages involves advantages and disadvantages. One of the disadvantages is that the proposed special-purpose interface conflicts with an often formulated requirement: "The messages accepted by the system must have the minimum of format and the maximum of approach to the natural way of dialogue between persons" (TAROUCO (1981)). The introduction of types of messages restricts the natural way of composing messages, by imposing formats and communication conventions.

However, there are some advantages to be considered. Each participant in a message exchange is enabled to imagine the conditions of message receipt, such as rules for message acceptance or levels of priority to give or to expect feedback. Furthermore, message exchange does not suffer from incompleteness or badly arranged pieces of the message body.

3. RESPONSIVENESS AS A DESIGN GOAL

3.1. PSYCHOLOGICAL BACKGROUND

The efficiency of problem solving in a team depends on the extent to which a problem solver is attentive to problems and suggestions of the other problem solvers in the team. The efficiency of message exchange depends on the extent of awareness of the participating individuals. This kind of attention towards the problem domain and the working conditions of the associated colleagues is called "responsiveness" in communication psychology (MEHRABIAN (1972)).

User-responsiveness as a design goal for the user interface of a CBMS means that the awareness of one another and the mutual reaction should be mediated by the CBMS. It is the author's experience that people sometimes tend to shower other people with information regardless of their information needs. This behaviour is contrary to responsiveness.

The submission of a message should be carried out with regard to the information requests of the recipient. Here, the sender may take the role of the addressee. There are good reasons to assume that people

are willing to bear the burden of information preparation in cooperative problem solving processes. WEEKS & CHAPANIS (1976) have observed that subjects solving cooperative problems spend a considerable amount of time in activities such as drafting messages for subsequent transmission.

Responsiveness in message exchange may help reduce the danger of an information explosion that could be produced by a message system that is used only for message transport purposes. Keeping responsiveness as an important design goal in mind, some related user requirements may be discussed.

3.2. SOME REQUIREMENTS

We are all familiar with the situation of someone who tries to phone someone else and who is surprised if not confused or embarrassed if he is unexpectedly confronted with an automatic answering device. In this situation the sender is temporarily hindered in preparing his message by the unexpected communication constraints.

The psychological reflection of "responsiveness" and the above mentioned example give reason in support of the following user-requirement:

> "The sender of a message should be enabled to get acquainted with the conditions of message receipt at the place of destination."

This requirement means that the sender should be assisted by the user-interface of a message system in recognizing the output conditions for the recipient.

Since team members expect that each of them is acting according to his role-defined responsibilities one may argue that each of them already has a clear conception of the conditions of message receipt at the various places of destination. This requirement is therefore partially satisfied by consensus on communication rules, division of labour, definition and treatment of message types.

However, this requirement is not met if some changes occur. To meet this case the requirement must be qualified:

> "The sender of a message should get acquainted with the change of conditions of message receipt at the place of destination."

For example, the sender should be informed as soon as possible if communication constraints have been changed or if the recipient is temporarily addressable at another place of destination or if someone is not currently available.

The requirements can be satisfied by a CBMS system component, called "assistant", which provides for the indication of changes and which guides the user in keeping to the rules of communication and the treatment of received messages.

It is not intended to "assist" communicative acts as described by CASHMAN & HOLT (1980) who implemented a monitor called MONSTR : "What is needed is a protocol-driven system. Such a system is given a description of the allowable paths of communication, the meaning of the messages sent over each path, and a description of the project's structure. It then constrains project communication to move along those paths and those paths alone" (p. 9). It seems to me

that people who are using such a system are in danger of having to communicate in a protocol-driven way, where the protocol is defined by the team manager in order to control more effectively the project. (I do not mean to imply this is the intention of the authors. However, I doubt that MONSTR is applicable in a software production environment.)

In the case of protocol-driven communication it is the manager who decides how the protocol is to be changed; implicitly he decides upon the rules of communication. However, there is another way to modify such rules, namely by "meta-communication" (WATZLAWICK et al. (1967)). In order to keep communication rules flexible, an additional user requirement for the design of the interface of a CBMS may be formulated as follows:

> "The message system should enable any user to communicate on different meta-levels and to switch between them."

Meta-communication is communication about "how" to communicate, i.e. communication about a rule of interaction between people. For computer mediated messaging one may consider the following levels of interaction:

- communication level: messages that are accepted by a CBMS;
- first level of meta-communication: rules for treating messages by sender or recipient;
- second level of meta-communication: rules for treating rules.

An important rule on the first level is that the sender should inform his counterpart with respect to his information needs. This rule is implicitly observed insofar as the sender responsively chooses the appropriate type of message. Another rule is that the message to be transmitted is split into two parts, one including an announcement and one including the message being announced. This rule is useful if the announced message is not suitable for a mailbox or if the recipient expects a message which is not yet ready or not yet available. It is also important to introduce a rule that applies if the CBMS is not available due to a system breakdown or other kinds of interruption. The user should be assisted in what to do in these exceptional cases.

On the second level of meta-communication people communicate about rules that may alter communication rules. A rule may be introduced to the effect that all members of a team are allowed to propose new rules of communication, for example, the proposal of a revised message type.

4. SUMMARY

A computer based message system (CBMS) should comprise more than elementary transport functions. If this electronic tool is integrated into the working environment of knowledge workers it should assist cooperation between them. To meet this goal, communication in a project team is taken as feedback control. The concept of feedback allows one to speak of the purpose of behaviour; hence, it also allows one to speak of the purpose of a CBMS in a project team. A requirement analysis for the user interface of a special-purpose CBMS should not start before feedback events and related types of messages have been carefully investigated. A major design goal for a CBMS is called "responsiveness". Requirements which contribute to this design goal are discussed, in particular, assisting the

co-ordination of team members by means of a CBMS and helping the team to move effectively towards a mutual goal. Protocol-driven versus rule-guided communication is briefly discussed. For a software production environment a rule-guided communication is suggested.

5. REFERENCES

BRUGGERE, T.H.: Software Engineering : Management, Personnel and Methodology. Proceedings of the 4th International Conference on Software Engineering, 1979, Munich, 361-368 (IEEE).

CASHMAN, P.M. and HOLT, A.W.: A communication-oriented approach to structuring the software maintenance environment. ACM SIGSOFT, Software Engineering Notes, Vol 5 No 1, 1980, 4-17.

DARLINGTON, J.; DZIDA, W. and HERDA, S.: The role of excursions in interactive systems. International Journal of Man-Machine Studies, to be published in May, 1981.

ENGLISH, H.B. and ENGLISH, Ava C.: A Comprehensive Dictionary of Psychological and Psycho-Analytic Terms. Longmans, New York, 1958.

MEHRABIAN, A.: Nonverbal Communication. Aldine-Atherton, Chicago, 1972.

MILSUM, J.H.: Mathematical introduction to general system dynamics. In: J.H. MILSUM (ed.): Positive Feedback - A General Systems Approach to Positive/Negative Feedback and Mutual Causality. Pergamon Press, New York, Toronto, 1968, 23-65.

TAROUCO, Liane: An experimental computer message system between universities in Brazil. Paper presented at the International Symposium on Computer Message Systems; this Proceedings Vol.

TEITELMAN, W.: A display oriented programmer's assistant. International Journal of Man-Machine Studies, 11, 1979, 157-187.

UHLIG, R.P.; FARBER, D.J. and BAIR, J.H.: The Office of the Future - Communication and Computers. North-Holland Publishing Company, Amsterdam, 1979.

WATZLAWICK, P.; BEAVIN, J.H. and JACKSON, D.D.: Pragmatics of Human Communication - A Study of Interactional Patterns, Pathologies, and Paradoxes. Norton, New York, 1967.

WEEKS, G.D. and CHAPANIS, A.: Cooperative versus conflictive problem solving in three telecommunication modes. Perceptual and Motor Skills, 42, 1976, 879-917.

COMPUTER CONFERENCING

Chairman - A. Danthine

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

THE EFFECT OF STRUCTURE, TASK, AND INDIVIDUAL ATTRIBUTES ON CONSENSUS IN COMPUTERIZED CONFERENCES

Starr Roxanne Hiltz, Murray Turoff, and Kenneth Johnson

Computerized Conferencing and Communications Center
New Jersey Institute of Technology
Newark, New Jersey
U.S.A.

One criterion of the effectiveness of a computer-mediated communication system for group decision-making is the ability of the group to agree on a decision at all. This paper presents data from two controlled experiments designed to answer the following questions:

1. How does a computerized conference differ from a face-to-face discussion in terms of the ability of a group to reach consensus?

2. How is this affected by the nature of the task?

3. How is it affected by the skills and characteristics of the participants, such as their previous use of computer terminals and their typing ability?

4. How is it affected by the specific structure used for the computer-mediated communication system? Specifically, can an explicitly structured human leadership role aid consensus in a computerized conference? Can a decision aid based on computer analysis and feedback of data related to the decision help?

5. What are the implications of the findings for the design and application of computer-mediated communication systems?

INTRODUCTION

Much of the early research on the social effects of computer-mediated communication involved attempts to make generalizations about the impact of this new medium. Thus, for instance, we find Johansen, Vallee, and Spangler (1979, pp. 180-181) summarizing a number of studies with the statement that "computer conferencing promotes equality and flexibility of roles in the communication situation" by enhancing candor of opinions and by helping to bring about greater equality of participation. On the basis of early pilot studies comparing face-to-face with computerized conferences, Hiltz and Turoff (1978, p. 124) conclude that, in computerized conferences, more opinions tend to be offered and asked for but that there is less explicit reaction to the opinions and suggestions of others, whether the reaction is that of agreement or disagreement. In terms of organizational impacts, Uhlig, Farber, and Bair (1979, p. 306) state that "collaboration of groups of persons, whether on a report or a complex decision, is accelerated by the speed of communication, including distribution and feedback." Among other early generalizations is one stating that it is more difficult for a group in a computerized conference or exchange of messages to reach total consensus on a decision than it would be for them to do so in a face-to-face meeting. As with the generalizations cited previously, this seems to be generally supported by further research. However, the second generation, so to speak, of research on computer-mediated communications has as its objective

a greater understanding of the conditions under which the general tendencies of the medium are stronger, weaker, or totally absent. Some of this research focuses on the structure or facilities of the computer-based communication system itself. Thus, for instance, current work at the Institute for the Future deals not with the social effects of a very simple, basic system (PLANET) but with the effects of adding three specific tools or structures to the basic conferencing program that are designed to support specific group tasks: "graphical communication...communication focused on the running of computer programs through its program workspace, and communication focused on the creation and editing of a document" (Lipinski, Spang, and Tydeman, 1980, p. 159).

Current work at the New Jersey Institute of Technology focuses on the development and evaluation of a variety of new capabilities for computer-based communication systems and on the discovery of interactions among task types, communications structures, and individual or group attributes that will in the future better enable us to choose the optimal systems to match variations in user-group characteristics and types of tasks or applications. The research program involves a combination of field trials and controlled experiments. This paper focuses on selected results of the first two of a projected series of three controlled experiments on group problem-solving via computer. A single dependent variable, the degree of group consensus on a decision, is measured for both experiments and is used to show how the effect of a computerized conferencing system on this aspect of the group process is related to variations in the structure of the computer-mediated communication system provided, the nature of the task being performed by the group, and attributes of the participants, such as their previous experience using computer terminals.

Computer-Mediated Communication: Generalizations and Variations

In computerized conferences, group members communicate by typing into and reading from a computer terminal rather than by speaking, listening, and exchanging nonverbal gestures. Each person types an entire entry without being interrupted, then receives communications that are waiting in the computer. The communication channel is, thus, missing many features of "normal" face-to-face communication, such as instantaneous receipt of communications and nonverbal cues (i.e., eye glance, facial expressions, tone of voice, and gestures). On the other hand, the presence of the computer in the communications loop provides some communications possibilities that are not available in a face-to-face meeting. For instance, all participants can think as long as they want, without being interrupted by others, before they make their comments. The participants can be on line at the same time in different locations ("synchronous" conferences or message exchange), or, more usually, they send and receive communications whenever they wish to, with the computer storing waiting communications until they are ready for them ("asynchronous" conferences). In synchronous exchanges, all can be typing at once, rather than having to take turns speaking. The printing-reading speed is faster than speaking-listening speed (30 characters per second was used in this experiment; 120 characters per second is not uncommon). The repeat key and special characters on the keyboard can be used easily to create linguistic and graphic effects, such as the use of a whole line of exclamation points or question marks for emphasis (see Carey, 1980).

The main varieties of computer-mediated communication are "messages" or "electronic mail," which stores and forwards discreet communications and may be thought of as replacing the internal memo, the letter, or the phone call; document and file transfer-systems such as NLS (now called Augment), which provide for communication through the transfer of files; and "conferences," which are oriented toward group communication by keeping a transcript of a single-subject discussion for a whole group and by providing features such as voting and markers showing where each participant is in the conversation. Though the "conference" structure is specifically designed to support group communication

and decision making so as to replace or augment face-to-face meetings, electronic mail systems of document and file transfer-systems can be used in the same manner, with the group members instead of the computer sorting and ordering the communi cations on a single problem or subject. In addition, many special structures or features can be created when the computer is in the communications loop. For instance, within a conference structure, a human leader or moderator can be given a very strong role. If there are data as well as purely qualitative communications involved, ranging from simple yes-no votes to large tables or files of information bearing on a decision, the computer can be used to serve as a decision support tool by analyzing, formatting, and feeding back the data to the group.

Peter and Trudy Johnson-Lenz (1980) have written about means of structuring computer-mediated communication as "groupware." They assert that

> For a group to use a computerized conferencing system effectively, it must have some explicit, intentional procedures to follow. These procedures set out the purpose of the group and its tasks, who can communicate with whom and when, how decisions are made and disagreements resolved, the sequence of activities to be used in accomplishing the task, and so forth. The procedures may be norms or rules enforced by the group, or they may include software enforcement. Such procedures constitute a communications structure, without which the group's work will be neither effective nor efficient. (Johnson-Lenz, 1980, p. 1)

The Dependent Variable: Consensus

There are two essential dimensions to a successful group decision: solution "quality" and "acceptance" or "consensus." The quality of a group decision may be assessed by comparing it to objective facts or expert opinions. However, if there is no group consensus or acceptance of a decision, then there may not be enough commitment to motivate successful implementation of a decision.

Total consensus is not necessarily a goal that is related to quality of decision. As Nixon (1979, p. 143) puts it in his summary of small group studies, "conformity and deviance can have either potentially functional or disfunctional consequences." However, consensus on a decision usually makes the group feel better about one another and about the decision.

DESIGN OF THE FIRST EXPERIMENT

This experiment is technically known as a two-by-two factorial design with repeated measures. This means that there were two independent variables, or "factors," and that each subject was measured in two different combinations of factors, or conditions.

Data were collected from sixteen groups of subjects who came to a laboratory at Upsala College; each of the groups engaged in two group decision-making conferences. The chief independent variable was mode of communication ("CC," or computerized conferencing," and "FtF," face-to-face). A variable of secondary interest was task type: Each group completed one task in one mode of communication, then a second using the other mode. Dependent variables included quality of decision and degree of consensus, the focus of this paper, as well as subjective satisfaction of participants and various measures of communications process. (Full details on the procedures and results for this experiment can be found in Hiltz, Johnson, Aronovitch, and Turoff, 1980.)

Another variable of secondary interest is the problem type. Much experimental literature indicates that the nature of the problem has a great deal to do with group performance. One type of problem that we used is the human relations case as developed by Bales. These are medium-complex judgmental problems that have no specific "correct" answer. The choice among possible options is related to basic values and beliefs about effective managerial style. The second type was a "scientific" ranking problem (requiring no specific expertise), which has a single correct solution plus measurable degrees of how nearly correct a group's answer may be. This ranking problem, "Lost in the Arctic," was adapted for administration over a conferencing system by permission of its originators (see Eady and Lafferty, 1975). It can be considered a "knowledge pooling" or "intellective" task (using the terminology of Laughlin and Adamopoulous, 1980).

The experiment took about four hours to run and involved a large number of instructions and actions by the experimenters and assistants, conditional upon the sequence to which the group was assigned. In the CC condition, all instructions were delivered through the computer. In the FtF condition, the subjects were seated at a round table, observed through one-way mirrors on each side, and recorded on a separate sound track for each participant.

Our initial hypotheses, based on a literature search and on our own pilot studies (Hiltz, Johnson, and Agle, 1978), were:

1. Computerized conferencing, as compared to face-to-face discussions, will probably result in more equal participation (confirmed).

2. This, in turn, is likely to lead to the generation of more ideas and suggestions on how to solve a problem (confirmed) and, thus, to a higher quality decision (not confirmed--the decisions were of equal quality).

3. CC groups are less likely than are FtF groups to reach total consensus in a given amount of time, since it is less likely that a single leader will emerge to push the group toward agreement. (The data on degree of consensus are presented later.)

A related factor is that the absence of nonverbal communications makes it much easier for a "deviant" group member to hold out against the other members of the group rather than go along with the group. In the pilot studies, there were no groups in the face-to-face condition in which a 4-1 split was maintained; the deviant always reluctantly went along. In the computerized conferencing condition, there were several instances of a stable, adamant 4-1 or 3-2 split, with the deviant(s) steadfastly holding that he or she did not agree with the rest of the members. Thus, we predicted that the computerized conferencing condition would be characterized, as compared with face-to-face discussions, by a lower probability of reaching a total consensus.

Ability to Reach Consensus

For the ranking problem, consensus was measured by using Kendall's coefficient of concordance for the five "final group rankings" reported by each individual in each group. This varies from 0 for no agreement to 1.00 for perfect agreement on the placing of the fifteen items ranked by the group. The results are shown in Table 1. There is a statistically significant difference in favor of face-to-face groups. However, the difference is not substantially very large; all CC groups reached a reasonable amount of agreement. Some of those groups that did not reach near-total agreement seem to have run out of time. Whereas all face-to-face groups completed their task within the ninety minutes allowed, many of the CC groups were cut off before they were able to finish. However, this is not the only factor. The computerized conference seems to provide little opportunity for a dominant leader to emerge and force a consensus, while at the same time it

fosters an environment that is psychologically and socially more conducive to allowing persons to refuse to go along with the group when they think their decisions are better than those of the rest of the group members.

An interesting sidelight is that all of the face-to-face groups apparently THOUGHT that they had reached total consensus. However, in half of the groups, when individual members were asked to report this agreed-upon decision in writing after the meeting, their versions of the decision were somewhat different. This is despite the fact that the participants usually wrote down the supposed decision on a list of the items they had with them in the conference room and later referred to it in reporting the decision.

There was a significant interaction between sex and mode of communication. In FtF, the women were more likely to change their opinions and go along with the men, thus helping to produce the apparent consensus. In CC, however, the women did not give in--they were no more likely to change their rankings than were the men. Our groups did not have any "token" females--that is, there were no situations in which a single female faced four males (see Kanter, 1977, for a discussion of the pressures on tokens). It is probable that there would have been an even larger difference in the degree of compliance of women between the face-to-face and the computerized condition if they had been tokens.

Table 1

Group Consensus on the Ranking Problem, by Medium of Communication

Face-to-Face	CC
.9897	.9974
1.00	.8626
.9886	.9031
1.00	.9857
.9943	.9671
.9989	.9811
1.00	.9737
1.00	.8077

Mann-Whitney U test
Ub = 0
p .01
Note: 1.00 means perfect consensus, all five participants on all fifteen items ranked.

The difference in ability to reach consensus was much greater for the seemingly simpler, but amorphous and value-laden, human relations problem (see Table 2). Consensus here was coded by simply looking at or listening to the final opinion given by each member and seeing if there was agreement. According to the transcripts, only one of the eight CC groups reached consensus on this problem, whereas all of the face-to-face groups reported reaching consensus.

We think that part of the difference is that announced consensus in the face-to-face groups may have, in reality, not been present. Unlike the procedure followed for the complex ranking problem, the members of the group were not required to state explicitly what the "group decision" was or whether they agreed with it. It is quite likely that, in at least some of the groups, there were persons who did not agree with the decision announced by a person playing a leadership role but who chose not to make their disagreement explicit.

However, most of the apparent differences are probably related to aspects of the nature of the two tasks and the structuring of the interaction processes used in the tasks.

In relation to quality of decisions for the Forest Ranger problem, as judged by experts, there was no difference in the types of decisions made by the face-to-face groups and the majority in the CC groups. However, the minorities in the CC condition were less likely to agree to take punitive actions, such as a pay cut, to correct the situation. The highest-rated solutions, as judged by the three personnel experts who rated the possible solutions, were made by the minorities in the CC condition who refused to agree with the solutions proposed by the majority. This has interesting implications for managerial applications of the medium.

Table 2

Inability of Computerized Conferencing Groups to Reach Consensus on a Judgmental Problem

Mode	Consensus	No Consensus	Total
Computerized	1	7	8
Face-to-Face	8	0	8
Total	9	7	16

Chi square = 3.06; p .05

The Importance of Task

We have seen that the results of comparing computerized conferencing with face-to-face discussions as far as ability to reach consensus is concerned depended somewhat on which of the two problems was being discussed. We initially thought of the problems as different, in the sense that the Forest Ranger is a simple human relations problem and that Lost in the Arctic is a complex, scientific-type task with a correct answer. However, even ignoring mode of discussion, there are other differences evident in the two problems. Lost in the Arctic, though a complex and somewhat difficult task, is more interesting and much more structured. It is clearer to the participants what they must do and easier for them to attack systematically and complete the problem.

Typically, a group attacked Arctic by comparing their fifteen initial rankings and by then picking out a subset of items near the top; then they agreed on what would be number one, then number two, etc. With Forest Ranger, it seemed to be much more difficult for a group to know where to begin and to know how to focus their discussion. Our feeling is that an unstructured judgmental problem needs strong human leadership to structure the discussion and decision-making process; computerized conferencing does not appear to facilitate the natural emergence of a leader.

STRUCTURING THE GROUP PROCESS: DESIGN OF THE SECOND EXPERIMENT

Two factors were chosen as forms of structuring interaction. One was the selection of a formal group leader, who had certain defined responsibilities. Groups in this "Human Leadership" (HL) condition were asked, at the end of their training and practice session, to rank-order the group members in terms of their ability to lead a group discussion. A weighted scoring of five points for first

choice, four points for second, etc. was used to calculate the group's votes. After their initial ranking for the Arctic problem, groups in the Human Leadership conditions were told whom they had chosen to lead the discussion and what the responsibilities of the leader would be. These responsibilities were to focus discussion, suggest specific ranking changes to reach consensus, and summarize progress. The control groups had a comparable task. They rank-ordered five candidates for President of the United States; there was no reporting of the group's choices. (See Hiltz, Turoff, and Johnson, forthcoming, for a complete report on this experiment.)

The second factor was the use of the computer to compile, analyze, and feed back to the groups information on the distributions of the rank orderings at various points in time. All groups received a table that was simply a list of the rankings of the members. The updated table was printed after initial ranking and before discussion began and then every ten to twenty minutes during the maximum of two hours that the groups had for discussion and reranking at will. The algorithm was that the table printed every twenty minutes by default, if it had not been generated sooner. After a table was printed, whenever a group member changed his or her rank order all rankings during the next ten minutes were collected and compiled, and the new updated table would then be printed.

In the "Computer Feedback" (CF) condition, a second table was generated (see Table 3). This listed the items in order of their mean ranking by all group members, showed the amount of agreement on each item, and reported two measures of amount of agreement so that the group could follow its progress toward consensus.

We thus had a two-by-two factorial design. There were six groups per condition, with five members per group. One of the conditions (No Leader, No Feedback) was comparable to the unstructured conferencing condition used in the first experiment.

The second experiment was a field experiment, with five staff members in a variety of organizations serving as subjects in what was termed a "participatory seminar". The organizations included Banker's Trust, Texas Instruments, and Chemical Abstracts, Inc., among others.

Table 3
Sample of Computer Feedback Table
(for the Practice Problem)

The overall group agreement is 56.8%.

.-Average Rank for This Item	.-Group Agreement	Item	User's Rank 901	902	903	904	905
1.2	84%	B Mousse	1	1	1	2	1
2.8	32%	E Strawberry	5	4	2	1	2
3.6	64%	C Pie	3	5	3	4	3
3.6	56%	D Cake	2	3	4	5	4
3.8	48%	A Crepes	4	2	5	3	5

Kendall's agreement coefficient is 0.464.

Table 4
Condition by Mean: Kendall's Coefficients

Condition	Kendall's
Human Leader, No Feedback	.980
No Leader, Feedback	.929
Human Leader, Feedback	.854
No Leader, No Feedback	.849

Analysis of Variance
Feedback- F = .35, p = .56 (NS)
Leadership- F = .50, p = .49 (NS)
Feedback by Leadership Interaction- F = 6.92, p = .02

Results

As shown in Table 4, there was a significant impact of human leadership or computer feedback on the ability of a group to reach consensus in a computerized conference. The effect was dependent on a strong interaction effect--as compared to an unstructured conference, either a human leader, alone, or the computer feedback tables, alone, aided consensus. However, in combination they canceled one another out and were no better than a conference structure without either aid. This somewhat anomalous result might have been different if the human leader had been given special training in the interpretation of the feedback tables so that he or she would not make incorrect interpretations that caused dissention in the group. However, we suspect that lack of skill in interpreting the tables was not the problem, but, rather, the attempt of several leaders to interject their biases into their suggestions for actions by the group to reach consensus was. One possibility is that, if both a decision aid and a human leader are present, the results of the analysis should go only to the leader so as not to undermine his/her authority if there is a preference for ignoring the data and enforcing one's own opinions. However, we suspect that such a procedure, though it would raise the consensus score, would also lower the quality of decision.

There was a strong correlation between typing ability and previous experience with computers and the ability of the group to reach consensus. The effect of skill is all the more striking because there is no significant relationship between these skills and the initial (prediscussion) degree of consensus.

Typing was measured on a four-point scale: hunt and peck, rough or casual typing, good typing (30 wpm, error-free), and excellent typing. Past use of computer terminals, for any kind of application, was self-reported as never, once or twice, three-ten times, or frequently.

The measure of correlation used is Pearson's. The square of this coefficient is the proportion of variance in the dependent variable explained by the independent variable. So, for instance, our correlation of .67 between previous experience with computer terminals and the amount of agreement among the final individual rankings can be interpreted to mean that 45% of the variance in the amount of consensus on final individual rankings can be predicted on the basis of previous use of computer terminals. In other words, previous individual experience is strongly related to the effectiveness of the use of computerized conferencing to reach consensus. The p is a measure of statistical significance.

Typing ability and previous experience with computer terminals are interrelated,

as would be expected (gamma = .36, p = .17).

The results are consistent with comparisons of our first-time users (experimental subjects) with more experienced users from the field trials. We find that amount of time on line is highly correlated with willingness to use CC for more complex communication tasks.

Table 5

Pearson's Correlation Coefficients
between Typing Ability and Previous Use of Computer Terminals
and Ability to Reach Consensus

Variable	Typing, Pearson's	p	Computer Terminal Experience, Pearson's	p
Initial Kendall's	-.09	.34	.11	.31
Group Kendall's	.40	.03	.56	.01
Final Individual Kendall's	.49	.01	.67	.00

SUMMARY

1. We have seen that the effectiveness of CC versus FtF as measured by the criterion of ability of a group to reach consensus depends somewhat on the nature of the task. Fairly well-structured tasks that involve some data or knowledge are more easily handled in computer-mediated communication, as compared to fairly amorphous tasks based on the exchange of highly subjectived opinions.

2. The particular structures or aids built into a computer-mediated communication system can have a significant effect on its ability to accomplish a task, which, in our experiments, was to reach consensus. Either an explicit structuring of a human leadership role or the provision of a computer analysis and feedback of data on the group's preference can aid consensus, as compared to free-form or unstructured discussions.

3. Individual characteristics, such as typing ability and previous use of computer terminals, are also highly related to the effectiveness of CC. In regard to the last factor, other studies indicate that typing ability and previous experience with computers or computer terminals do not significantly affect eventual acceptance of and facility with CC, but they are factors that will tend to decrease learning time and increase initial effectiveness. A practical implication of this finding is that a group should have some facility with a computer-mediated communication system before it is put under pressure to use that system to resolve a pressing problem.

IMPLICATIONS

Although most of the computer-mediated communication systems going into place today are viewed as replacements or replications of the internal mail system, this is not where the potentially largest payoff for computer-mediated communication systems lies, in our opinion. An electronic mail or message system can be cost effective as a replacement for the paper mails (Hiltz and Turoff, 1978; Turoff, 1971, 1972). However, this ignores the opportunity for dealing, in the new and unique ways made possible by the technology, with the problems the organization faces. An objective for such systems should be to

facilitate geographically dispersed groups to act optimally as teams to grapple with the complex problems facing the organization. The real opportunity for an organization arises when it can tap the right people to deal with a problem regardless of their location and can minimize through computer-mediated communications systems the need for travel and face-to-face meetings.

It is entirely possible that some problems will receive better treatment in this form of communication than they would if the groups were given the opportunity to meet face-to-face. We feel that this can be particularly true when the objective of the group is to develop for the decision makers a range of well-thought-out alternatives. This is, of course, the same idea that underlies the Delphi method and the use of that technique today by a significant number of organizations. The Delphi method is basically a special communication structure designed around the type of problem being dealt with by the group. Many of the successful Delphi structures implemented through paper-and-pencil iterative surveys represent unexplored computer-mediated communication-system designs. However, the introduction of the computer raises a large number of design issues that never occurred in the manual Delphi approach. Our first two experiments serve to explore only a very few among the many variables and issues that need to be studied.

The first issue is the conflict that arises when one designs features for the benefit of the individual's objective as opposed to the objectives of the group of which the individual may be a member. The issue of consensus and of fostering consensus among a group is a case in point. Systems designed with the individual primarily in mind may lessen the ability of a group to reach consensus more easily by computer-based communications. Inherently, the presumptions underlying a message-system design are usually based upon considerations of the individual user, not the group that may represent a task team.

Another open issue is the degree to which human facilitation roles can be augmented or substituted for by a computerized process. On the EIES system, there is specific software to support numerous facilitative roles that have evolved over time. We expect that this will be true for any system operating in an organization. People who have leadership, coordination, or facilitation responsibilities will desire features to improve their ability to carry out those functions. As a result, not every individual user will be equal in such systems, and one can expect the emergence over time of an organizational or social structure.

The third issue is the degree to which a complex problem can be structured by the communication process. The Forest Ranger problem, though value-laden and ill-structured by nature, lends itself to a variety of communication structures that could have been used to focus the group either on generating alternatives or on pushing through a consensus. We chose in this first experiment not to do so, since we were more interested in identifying the variability between a face-to-face meeting and a straightforward computerized conference. Most decision support systems today focus on the use of data analyses and data bases to provide the decision support functions. As a result, it is admitted by the writers in the field that decision support systems are largely for problems that can be semi-structured (Keen and Morton, 1978). Unfortunately, many problems facing the organization are ill-structured (Churchman, 1971). Our hypothesis is that many ill-structured problems can be made semi-structured by content-free communication structures made possible by putting human communications through the computer. Some examples of this already exist on the EIES system in the form of special structures to facilitate a group's joint exploration of ill-structured problems.

The best example of this, however, is still represented by the EMISARI system designed for the crisis management environment (Hiltz and Turoff, 1978; Kupperman and Wilcox, 1972).

In sum, we feel that the future of human communication via computer lies in the design of communication structures to fit alternative applications. However, we need a better balance between actual applications and careful experimentation than has been evident to date. Our results in this experiment are counter to the tradition of information systems, which assumes the more facilities provided, the better.

REFERENCES

[1] Bales, R.F., How People Interact in Conferences, Scientific American, 1955, 192, 3, 31-35.

[2] Carey, J., Paralanguage in Computer Mediated Communication, Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics and Parasession on Topics in Interactive Discourse, Philadelphia, Pennsylvania, June 19-22, 1980, 67-69.

[3] Churchman, C.W., The Design of Inquiring Systems, Basic Books, 1971.

[4] Eady, P.M. and Lafferty, J.C., The Subarctic Survival Situation, Plymouth, Michigan: Experiential Learning Methods, 1975.

[5] Hiltz, S.R., Johnson, K. and Agle, G., Replicating Bales' Problem Solving Experiments on a Computerized Conference: A Pilot Study, Newark, New Jersey, Computerized Conferencing and Communications Center, N.J.I.T., Research Report #2, 1978.

[6] Hiltz, S.R., Johnson, K., Aronovitch, C. and Turoff, M., Face to Face vs. Computerized Conferences: A Controlled Experiment, Newark, New Jersey, Computerized Conferencing and Communications Center, N.J.I.T., Research Report #12, 1980.

[7] Hiltz, S.R. and Turoff, M., The Network Nation: Human Communication via Computer, Reading, Massachusetts: Addison Wesley Advanced Book Program, 1978.

[8] Hiltz, S.R., Turoff, M. and Johnson, K., The Effects of Human Leadership and Decision Support Feedback on Group Problem Solving in Computerized Conferences: A Controlled Experiment, Newark, New Jersey, Computerized Conferencing and Communications Center, N.J.I.T., forthcoming.

[9] Johansen, R., Vallee, J and Spangler, K., Electronic Meetings: Technical Alternatives and Social Choices, Reading, Massachusetts, Addison Wesley, 1979.

[10] Johnson-Lenz, Peter and Trudy, Consider the Groupware: Design and Group Process Impacts on Communication in the Electronic Medium, in Hiltz, Starr Roxanne, and Kerr, Elaine, (eds.), Studies of Computer Mediated Communication Systems: A Synthesis of Findings (draft of final report to the National Science Foundation), 1980.

[11] Kanter, Rosabeth Moss.

[12] Men and Women of the Corporation, New York, Basic Books, 1977.

[13] Keen, P.G.W. and Morton, M.S.S., Decision Support Systems: An Organizational Perspective, Reading, Massachusetts, Addison Wesley, 1978.

[14] Kupperman, R.H. and Wilcox, R.H., EMISARI - An On-Line Management System in a Dynamic Environment, Proceedings of the 1st International Conference on Computer Communications, 1972.

[15] Laughlin, P.R. and Adamopoulos, J., Social Combination Processes and Individual Learning for Six-Person Cooperative Groups in an Intellective Task, Journal of Personality and Social Psychology, 1980, 38, 941-947.

[16] Lipinski, Hubert, Spang, Sarah and Tydeman, John, Supporting Task Focussed Communication, in Benenfeld, Alan, R., and Kazlauskas, Edward John (eds.), Communicating Information: Proceedings of the 43rd ASIS Annual Meeting, White Plains, New York: Knowledge Industry Publications, Inc., 1980.

[17] Nixon, H.L., II-The Small Group, Englewood Cliffs, New Jersey: Prentice-Hall, 1979.

[18] Turoff, M., Delphi and Its Potential Impact on Information Systems, Proceedings of the AFIPS Fall Joint Computer Conference, Volume 39, AFIPS Press, 1971.

[19] Turoff, M., Partyline and Discussion: Computerized Conferencing Systems, Proceedings of the 1st International Conference on Computer Communications, 1972.

[20] Uhlig, R.P., Farber, D.J. and Bair, J.H., The Office of the Future, North Holland, 1979.

*The research reported here was supported by a grant from the National Science Foundation (NSF MCS-00519). The opinions and findings are solely those of the authors and do not necessarily represent those of the National Science Foundation.

We are grateful to Charles Aronovitch for his participation in all phases of "experiment 1," to Julian Scher and Peter and Trudy Johnson-Lenz for their contributions to the design of the experiments, to John Howell, James Whitescarver, and Tom Moulton for their software development support, and to our research assistants, Joanne Garofalo, Keith Anderson, Christine Naegle, Ned O'Donnell, Dorothy Preston, Karen Winters, and Stacy Simon, for their dedicated efforts in carrying out the experiments and in coding questionnaires. We also wish to thank Nancy Rabke, who helped with the analysis of results as well as with the other phases of the experiments. The manuscript was edited by Diane Price. Robert Bales and Experiential Learning Systems were most cooperative in making experimental materials available for adaptation for these experiments.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

COMPUTERIZED MESSAGE SENDING AND TELECONFERENCING IN AN INTERNATIONAL ENVIRONMENT - PRESENT AND FUTURE

Istvan Sebestyen

International Institute for Applied Systems Analysis (IIASA)
Laxenburg, Austria

Computerized message sending and teleconferencing techniques are presently widely used in an international environment. This paper describes the present status (experience, problems, solutions) of electronic message exchange activities at an international research organization, the International Institute for Applied Systems Analysis (IIASA), and gives a short outline of some of the future prospects in this field.

1. INTRODUCTION

Computerized message sending and teleconferencing are closely related techniques of electronic information exchange. In fact, in many systems, both techniques are implemented at the same time, computerized teleconferencing being regarded as a special type of message sending. In this paper, therefore, it is not intended to draw a clear distinction between these two functions.

Teleconferencing is a term which has different meanings. Here, computerized teleconferencing is used to refer to computerized message sending and storage systems which enable text communication by terminals and computers through a computer network among participants in widely distributed geographical locations. The basic principle of computerized message sending and teleconferencing is the following: a so-called "mailbox" is assigned to each user or conference participant; messages or comments on the conference made by any user of the system are automatically put into the "mailboxes" of the other users; these messages are then "picked up" individually - but, of course, at different times. One characteristic of these teleconferencing systems is that the exchange of messages does not usually have the fairly limited duration - say, two hours - typical of "real" conferences: computerized teleconferences on certain subjects last, as a rule, for several weeks, or even months. Another characteristic of these systems is the "mailbox" principle, which brings alongside both advantages and disadvantages. One disadvantage is that the message recipients - normally the passive side, if one considers the traditional communication media, such as mail, cable, telex, telephone - now have to assume an active role in order to "pick up" the messages or conference comments from the mailbox. If access to the mailboxes is not gained regularly, senders of messages/conference comments often become frustrated and lose interest in using the system. However, an advantage of the mailbox principle is that only the "geographical" location of the "mailbox" is fixed, in the sense that it resides within a particular host computer. The recipient in the system may access the mailbox through the telecommunication network from any location he desires.

A computerized message sending and teleconferencing system can be regarded as a kind of new telecommunications medium which does not, however, replace the "old" well-known communication media; rather it adds a new dimension to them. This can be advantageous under certain circumstances, e.g., when documents are drafted by

a group of authors in different and remote geographical locations. As communication media, computerized message sending and teleconferencing systems are perhaps most closely related to telex-type systems, the basic difference between them being the "mailbox" principle of the former, and the fact that it also provides additional services to those of the basic telex-type functions (i.e., transmission of text to selected addressees over long distances through telex channels), such as:

-- text preparation and modification
-- automatic message filing
-- database-like message retrieval
-- management of text or message flow: keeping a record of incoming and outgoing messages
-- special teleconferencing services (voting, handling of questionnaires, "moderator" functions, and so on).

It is, therefore, possible to make the following - rather oversimplified - statement: a computerized message sending and teleconferencing system is the combination of an "upgraded", more complicated and reliable high-speed telex system, and a so-called office automation system. The latter type of system - based mainly on minicomputers - is currently expanding worldwide.

The function of the "upgraded" telex service is carried out by computer networks, such as the TYMNET or TELENET packet-switching networks. It should be added that computerized message sending and teleconferencing can now be regarded as among the most important applications of computer networks. The fact that not all PTTs sufficiently appreciate this function of computer networks should also be mentioned. For example, the British PTT only recently allowed selected British users to use the EIES teleconferencing system [1], provided they made a formal application to the British PTT.

Technically, such message sending and teleconferencing systems are built around computer programs and files located on any computer connected to the networks as a so-called "host". Users of such systems - namely conference participants - can access the network via terminals, such as alphanumerical displays, teletypes or even computers, and having established links to these "hosts", the users gain access to the computer programs and files used for teleconferencing. From a design point of view, it is easiest to put all teleconferencing computer program components and data files ("mailboxes" and the database containing messages) onto one central host computer in order to gain wide terminal access throughout the computer network. This solution has been chosen by the majority of the presently running message sending and teleconferencing systems, such as the EIES (Electronic Information Exchange System) of the New Jersey Institute of Technology, or the PLANET and NOTEPAD systems of the INFOMEDIA corporation [2], both located in the USA. One major disadvantage of such systems is that all the functions of the system are carried out centrally, which means, of course that, if the central computer ("host") is not operational (for instance, due to a technical failure), then access to the messages and programs is not possible. Furthermore, centralization of the normal "office automation functions" (such as text preparation and modification) into a central computer system leads to increased data traffic between terminals and the host, and thus to an increase in total data communication rates.

At present, the "second generation" of computerized message sending and teleconferencing systems is under development (such as the KOMEX system [3] of the Gesellschaft fuer Mathematik und Datenverarbeitung in the FRG). These systems distribute the above mentioned functions among various computer systems in the computer network. In such systems, the datafiles (such as "mailboxes" and the databases containing messages), and also all those functions (such as text editing

and text formatting) that can be performed near to the location of the user, are first of all distributed. One characteristic of these systems is that they are more resistant to any kind of technical failure, since in such cases, only a part of the total system "goes down" while the rest of the system, running on other computers, can continue to function and even take over partly the job of the malfunctioning computer. The telecommunication charges incurred as a result of the increased amount of local work will be less than those relating to systems of the first generation.

2. TELECONFERENCING IN AN INTERNATIONAL SETTING - THE IIASA CASE

The International Institute for Applied Systems Analysis [4], located in Laxenburg near Vienna (Austria), is supported by 17 so-called National Member Organizations (NMOs) in Western and Eastern European and North American countries. These countries are thus widely distributed throughout the Globe, thereby "providing ideal conditions" for teleconferencing activities. In order to coordinate the Institute's research activities with those of its collaborators, i.e., National Institutes or individual scientists, it is necessary to have a broad communication system between IIASA and the outside world. This function is partly based on traditional communication media (such as mail, telexes and telephones) which - equally traditionally - work badly, as a result of the "reliable service" provided by the PTTs. The above mentioned communication requirements of the Institute are also partly fulfilled by computers and computer networks. At IIASA, there are different types of computer communications: e.g., remote usage of computational programs, exchange of data files for model runs and text communication (such as message sending and teleconferencing).

Different systems for text communication have been used throughout the history of the Institute, such as the FORUM system of the Institute for the Future (IFF) [5], the PLANET system on the computer of the USGS (US Geological Survey) in Denver, Colorado, and of course the EIES system of NJIT in Newark, USA. A simplified - EIES-type - teleconferencing system called TELECENTER [6] is also in operation on IIASA's in-house PDP 11/70 computer. The Institute also works in close cooperation with the "Gesellschaft fuer Mathematik und Datenverarbeitung" (GMD) in Sankt Augustin near Bonn, FRG. There are plans to use GMD's distributed computer conferencing system, KOMEX, for a field experiment at IIASA.

As mentioned above, the EIES system of NJIT is presently the most frequently used system at IIASA for teleconferencing purposes. The current teleconferencing activities at IIASA are not always of the "classical" type, which means that the various small groups of people communicating with one another rather seldom solve problems by computer teleconference-based discussion, as was the original idea of teleconferencing: IIASA, rather, exploits the fast and reliable telecommunication and office automation facilities of such systems.

2.1 The Following Types of Message Sending and Teleconferencing Application proved to be most successful at IIASA:

a) Simple Message Sending

This "telex-type" function, supported by smart office automation facilities, such as text editing (entering and modifying text) and organizing, storing and filing information, proves to be most useful when access through other means of communication (telephone, telex) is at times more difficult or slower. The computer has been used for message sending, especially for our connections to Italy, USA, the Soviet Union and Hungary, at those times when no quick connection could be established through the normal channels of telephone and telex. We are fully aware that this kind of usage of remote

computer lines is not very favorably looked upon by the PTTs, but sometimes we are forced to do this. We would promise to discontinue this type of usage of computer lines if the PTTs were to provide us with a reliable and quick telephone and telex service to these countries.

b) Joint Manuscript Writing

The writing of joint manuscripts by a small group of geographically dispersed people proved to be one of the most successful message sending and teleconferencing applications. The quick and reliable message sending capability of computer networks, and the built-in office automation capabilities of the central computer allow the joint manuscript writing process to be most effective. In the summer of 1979, for instance, a joint "Working Paper" of around ten pages was prepared by A. Gruebler of IIASA and R. Sheldon of EWRSI in Honolulu (Hawaii) within two weeks. The Survey Project at IIASA is at present broadly discussing manuscripts, as well as transmitting and modifying sections of them for chapters of the "Handbook of Applied Systems Analysis" and for volumes of the "International Series on Applied Systems Analysis" published by John Wiley & Sons. The "office automation" functions that proved to be most successful when preparing joint manuscripts were text editing and text formatting (formatting material which has been entered or modified by text editing).

c) Administrative Management of Joint Projects

Very positive experience has been gained at the Institute in the field of administrative management of joint projects by computerized teleconferencing systems. Not only do these systems allow quick information exchange among many participants, but also through the capabilities of these systems, e.g., filing, keeping records of information (such as letters, protocols, text, documents, addresses, etc.) and their more specialized "office automation" functions (such as maintaining appointment calendars, action notices, scheduling aids, etc.), a set of useful management tools has been established in addition to the basic message sending functions.

The decision-making process relating to projects carried out by International Organizations is generally very long as these decisions, being made not only by one person or one central organ, are usually of a very democratic nature; therefore it takes much longer to arrive at final decisions. The wide geographical distribution of the decision makers leads to further long delays. This is not meant to be a complaint, but rather a fact. For instance, in order to initiate a UNDP (United Nations Development Program) project, two to three years of hard preparatory work is necessary, and agreement has to be reached among the UNDP, UN specialized agencies (e.g., UNESCO), the UNDP country representative and the governments and governmental agencies involved. We believe that the management of such projects can be improved significantly with the support of computerized message sending and teleconferencing systems.

One successful experiment of this kind using teleconferencing was the preparation of the large IIASA-RSI Conference [7] early in 1979. For this experiment, scientists from EWRSI, Hawaii and from our Resources and Environment Area, using a teleconferencing system on the computer at the US Geological Survey in Denver, Colorado, made all the necessary preparatory managerial arrangements for a large conference on mineral resources, which was later held at IIASA with great success. Similar projects of this kind are at present being carried out.

One interesting application of teleconferencing presently used at IIASA is associated with the operation of the Institute's external computer links. The TELECENTER teleconferencing system on the inhouse PDP 11/70 computer is

used for exchanging and filing messages relating to the operation of IIASA's network. A special conference on TELECENTER has been set up for this purpose and all operators of IIASA's network nodes, including Moscow, Prague, Budapest, and IIASA itself are participants in this conference. The exchange of daily messages, such as: "IIASA's PDP computer will not be operational on the afternoon of Friday, 13 January", or the scheduling of remote computer resources for the following week, are typical applications.

d) Teleconferencing in the "Classical" Sense

Surprisingly, our experience of "classical" teleconferencing activities, i.e., discussions by a group of geographically dispersed experts on a certain problem and contributions by these decision-makers toward its solution, is rather limited in comparison with other types of message exchange applications. One of the reasons for this is that the "telecommunications infrastructure" at IIASA is still expanding. It is hoped, however, that the number of future teleconferencing activities of a "classical" nature will increase.

2.2 Problems connected with, and Deficiencies in, the Usage of the Present Teleconferencing Systems

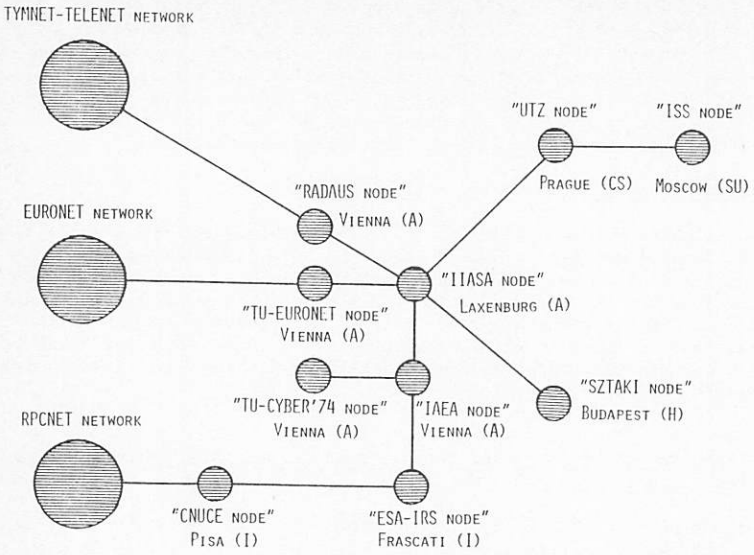
In the previous paragraphs, the advantages of computerized message sending and teleconferencing systems were shown. Although the picture is in general optimistic, the various problems we encountered when using these systems should be pointed out at this stage:

a) Lack of Standards

Lack of standardization of the routine functions relating to message sending and teleconferencing systems has led to the necessity of learning as many different text editing, text formatting, message sending, and text retrieval languages as there are systems to be used. One of my colleagues recently complained that he was "sick and tired" of learning five different text editing, text formatting and message sending systems with the same functions. A lack of standards also makes it more difficult to link local text editing systems to remote message sending and teleconferencing systems. No standard for file transfer exists, which should be one of the most essential and important functions. Much has still to be done in this field.

b) Insufficient Geographical Coverage of Computer Networks

One of the most serious problems IIASA faces with regard to teleconferencing is insufficient geographical coverage of the computer networks available at present for this purpose. This is the reason why IIASA does not at present operate its own teleconferencing system worldwide, using mainly the EIES teleconferencing system of the NJIT in the USA. Today the widest geographical coverage of computer networks is obtained if a US host computer with appropriate teleconferencing software is chosen. In this case, 14 out of the 17 National Member Organizations of IIASA may access the host. Under this arrangement, North America, Japan and Western Europe are fully covered; Eastern Europe may have access to the Austrian node of TYMNET/TELENET through the so-called "IIASA gateway". The necessary arrangements with the selected host, the telecommunication carrier and IIASA must, of course, be arranged previously. It is hoped that Eastern European coverage through existing lines to the USSR, Hungary and Czechoslovakia, and through the establishment of the planned new computer line to Bulgaria, will bring almost all IIASA's National Member Organizations together in a common teleconferencing system (Figure 1).



IIASA'S EXTERNAL COMPUTER LINKS
FIGURE 1.

IIASA also hopes to run its own "Teleconferencing Center". For this purpose, we are now successfully experimenting with teleconferencing software systems on the in-house PDP computers, such as TELECENTER, and are planning to carry out field experiments with the GMD teleconferencing system, KOMEX. However, the geographical coverage of a host computer at IIASA in Laxenburg is at present limited: Eastern Europe, with its existing (Budapest, Moscow, Prague) and future (Sofia) computer lines, is more or less covered, or will be covered; North America is also covered; however, Western Europe is not, since as a result of the PTT's restrictions, access from a European terminal to a European host via TYMNET/TELENET is not permitted, although technically feasible, and although the IIASA computer center is a host on the TYMNET/TELENET network. In order to cover a substantial part of Western Europe, it is necessary to wait until there is the possibility of Austrian host connection to EURONET, the computer network of the European Communities. This, we hope, will take place in the summer of 1982. Therefore if TELECENTER is used, only 8 out of IIASA's 17 member countries will be able to have access to our inhouse computer. A further problem is also the fact that EURONET, in its present phase, is restricted to database applications; it is hoped, however, that within the framework of an experiment, IIASA will obtain the permission of the Communities to use EURONET for teleconferencing purposes. It is generally to be expected that, in the next couple of years, the situation with regard to insufficient geographical coverage will improve.

c) Lack of Appropriate Teleconferencing Software

As mentioned earlier, there are many systems available and many under development. Most of the systems with good message sending capabilities have only primitive "office automation" facilities, or vice versa. At present, in our opinion, the most advanced operational teleconferencing software systems

accessible from IIASA are the EIES system of NJIT and the NOTEPAD system of the Infomedia corporation. Nonetheless, there is ample room even for their improvement: first and foremost, there is a need for more decentralization and better file transfer facilities. It is hoped that the "second generation computer software mentioned earlier will fill these gaps.

d) Problems of Man-Computer Communication

As IIASA is a scientific institution, most scientists using the computer to communicate are already experienced computer users. Thus the "conference participants'" fear of the computer is not really an IIASA problem, but it definitely is so in a less scientific environment. The scientists at IIASA using, for example, the EIES system have to be familiar with the UNIX operating system of the in-house PDP computer, or the DOST system of IIASA's TPA 70 computer [8], since connection to the TYMNET/TELENET network can only be made from IIASA through these computers. The command language of the TYMNET/TELENET network then has to be mastered in addition; this is not very complicated, but still a new "language environment". On top of the two languages previously mentioned, the final step is the mastery of the complex command language of the EIES host system in Newark. For users who are not experienced, the need to acquire this basic computer knowledge is certainly a burden.

e) Legal Problems

Nowadays computerized message sending and teleconferencing systems have become a reality whether some of the PTTs like it or not. Their legal status is still, however, under discussion. Do they provide services which interfere with the interests and monopoly of the PTTs? And, if so, can and should the service simply be forbidden? If their usage is not permitted, who will control it, and how? There is probably no way of controlling this.

At present the attitude of the PTTs toward computerized message sending and teleconferencing varies considerably: some of them seemingly take no notice of it, and have no objection to these kinds of activities; others, like the British Post Office are more strict about it. It seems to be most difficult to filter, out of the mass of different sorts of computerized data traffic flowing over computer networks, those parts which carry the text of messages or conference comments. In some countries, however, computer network-based message sending services are receiving the "blessing" of the telecommunication authorities. The US Federal Communication Commission (FCC) has just recently licenced the message sending services of the GTE TELENET network (TELEMAIL) [9], and of the TYMNET network (ONTYME-II). Both services offer message sending, text editing and other facilities, partly similar to those of the well-known teleconferencing systems. It is probably only a matter of time until these network operating companies will include teleconferencing and other "office automation" facilities in their normal services. It is my personal belief that, sooner or later, the different PTTs will themselves offer high quality computerized message sending and teleconferencing facilities on a large scale over their public networks, and this might replace the role of the present "private host"-based teleconferencing systems.

International teleconferencing is, from a legal point of view, even more complicated: the usual questions relating to Transborder Data Flow, such as interdependency, vulnerability, privacy, etc., only complicate the legal framework around this new medium. It is the author's personal belief that - for example - if a foreign host-based teleconferencing system were to be used for a conference of particular importance in a given country, this would be viewed with great concern by many governments. It is also not quite clear in what form these teleconferencing comments can be processed further. Is there

a copyright on these comments and, if so, who would be the copyright holder: the author of the comment or the conference moderator? These questions are not of the purely theoretical type, but problems that have been encountered at IIASA while using these systems. These problems will certainly be solved, but it will take a long time and much effort.

2.3 Summary of Current Teleconferencing Activities in an International Environment

In the previous chapters, the pros and cons of international message sending and teleconferencing have been outlined, based mainly on the experience gained at the International Institute for Applied Systems Analysis.

The capability of this new communication medium to improve telecommunications between the outside world and IIASA provides solid ground on which to build future computer-based message sending and teleconferencing activities at the Institute, especially in our international setting. Through this new medium, we have already established new links to geographical areas which - "thanks to" the PTTs - are less reliably accessible through normal telephone or telex connections, and the sending of mail also takes too long.

An additional capability of computerized message sending and teleconferencing systems, i.e., to combine the advantages of computer-supported "office automation" systems and telecommunications systems, adds new horizons to this communication medium.

Computerized message sending and teleconferencing activities will certainly achieve a breakthrough in this decade. They will be widely used by the international organizations for promoting cooperation between nations. The present technical and legal difficulties will be solved, hopefully soon; however, in order to achieve this, the PTTs and similar bodies will have to carefully assess the role of these systems.

3. PROSPECTIVES FOR MESSAGE SENDING AND TELECONFERENCING VIA COMPUTER

It has also been recognized, however, that the present types of computerized message sending and teleconferencing systems, such as EIES or NOTEPAD, represent only first generation systems in the area of computerized text communication media. As mentioned earlier, it is well known that serious efforts are currently been undertaken to improve the present systems and to develop second generation systems for the purpose of satisfying the needs of the next few years to come. However, also in a long term perspective, a revolution of the present telecommunication media can be expected. Signs of this revolution are already with us now, for example:

-- in the field of telex communication, new telex network installations are already being equipped with computer-based switching nodes. In addition, a new type of telex station, with built-in text processors and secondary (e.g., discette, cassette) storage capacity are available on the market, as the old telex system gradually becomes a more computerized medium. Teletex, the "fast-telex" recently standardized by CCITT enables A-4 pages of "correspondence quality" to be transmitted in about 10 seconds. In addition to message handling, Textstations - the terminal equipment for Teletex - can be used, in the same way as normal typewriters with text storage capacity and built-in intelligence, for text editing and text formatting. It is expected that this service will be introduced and interlinked with the telex networks on a worldwide basis in the first half of the eighties.

-- the "good old", but unfortunately increasingly expensive and unreliable, mail postal service is also moving in the direction of computerized "mass mail services". For example, the United States Postal Service (USPS) has launched an experiment known as EMSS (Electronic Message Service System). A limited version of this system linking up to 10 post offices within the United States is planned for 1981; a nationwide system should become operational in 1990. Conventional letter mail would be fed into high speed facsimile equipment and would then be routed via terrestrial or satellite circuit to a destination post office. Here the message would be converted back to hard copy for delivery by the traditional postman. In addition to this service, plans are being made for on-line input and output via computer, magnetic tape units, and keyboard terminals and printers to be located on sender and receiver premises. This last function resembles those of the present computerized message sending systems. In Europe, the French PTT has similar plans: preparations are now underway for experiments on the international level through satellite.

-- new types of communication networks are currently being developed. By way of example, in the US three new networks are now under development and will become available in the next couple of years:

- o SBS - Satellite Business Systems, a total communications network proposed by a joint venture of IBM, Comsat, and Aetna Life and Casualty
- o XTEN - Xerox Telecommunications Network
- o ACS - Advanced Communication Service, the data communications networks planned by the Bell System

All of these planned networks will support large scale electronic message sending and teleconferencing in one of several ways: IBM projects that 75% of the traffic on their SBS system will carry voice and 25% will be used for facsimile, multipoint distribution of digital data including document distribution, and video teleconferencing.

XTEN of Xerox will be primarily an electronic message service, but in addition, will support new applications such as document processing and distribution, data transmission and computerized and video teleconferencing.

ACS of Bell (AT & T) is closely related to the present US packet switched computer networks TYMNET and TELENET. It is to be implemented using minicomputers at the network nodes to add intelligence to existing AT & T intercity digital facilities. Among the facilities which are to be offered by ACS, there is one allowing users to customize their virtual subnetworks to their particular way of data handling. Other facilities include message preparation aids, editing capabilities, and message handling functions, such as storage and retrieval of messages, sending messages, building messages for records, etc. Message sending systems similar to ONTYME-II and TELEMAIL could easily be installed.

Similar networks are being discussed or are in preparation in other countries too. One must recognize that all of these systems support electronic message sending functions at a high level.

-- Videotex [10], another new telecommunication medium, is said to be the most radical invention in consumer electronics since television itself. Through Videotex, electronic message sending systems will go "public"; not only data base accessing, but also electronic messaging will become publicly available by means of inexpensive terminal devices, provided the PTTs allow it. In the basic versions, such as PRESTEL of the British Post Office, this is achieved

by transforming the home television receiver into a home information retrieval terminal, and important for us, into a home telex-electronic mail system. In the French (TELETEL) version, inexpensive facsimile terminals are being planned in addition for a mass consumer market. All types of "terminals" will provide subscribers to telephone and Videotex services with easy mass access through normal telephone lines to data bases located on PTT and, in some countries, on non-PTT computers which contain information about daily life or which are of general public interest. The terminals will also allow electronic mail messages to be sent from one terminal to another by an electronic computer-based mailbox system provided by the PTTs.

These seemingly independent developments mentioned above will certainly lead to a more advanced communications age, with broad overlapping of the functions of the various telecommunications media. When this stage is reached, the present "cottage-industry-like" message sending and telecommunications activities will be phased out and will be replaced by the "mass production methods" of the future communication networks.

What effect will these new developments in telecommunication media have on IIASA's electronic message exchange activities?

The telecommunication environment will not remain untouched by the developments outlined above. In Austria - the host country of the Institute - Teletex will be introduced in 1982. It is also of significance that the DATEX circuit switching network owned by the domestic PTT already provides 300 baud asynchronous connections; 2400, 4800 and 9600 synchronous DATEX channels will also be available later this year or in early 1982. The national DATEX-P packet switching service will also become operational in 1982. Moreover, the Austrian DATEX network will be linked to foreign PTT networks, such as the West German DATEX networks. This is expected to happen during the summer of this year.

The Austrian interactive Videotex system, "Bildschirmtext", will start an experimental service with 300 subscribers on March 1, 1981. The British Post Office started a market trial of its "PRESTEL International" service in 1980 in certain selected countries; an experimental worldwide service is planned, starting from the summer of 1981. It is foreseen that the "PRESTEL International Host" in England will be made available from Austria through the EURONET network. IIASA proposes to establish links to both PRESTEL systems. The advantage to IIASA of using PRESTEL International would be its broader geographical coverage, whereas the Austrian "Bildschirmtext" system, being closer to IIASA, would be cheaper and more suitable for in-house purposes.

It is hoped that a certain amount of the computerized information exchange traffic will be handled by both systems. IIASA, in its role as an "Information Provider" (to quote the Videotex jargon), plans to set up pages for wide public access on IIASA's current activities in general and also some pages for IIASA's "closed user groups", comprising the Institute's National Member Organizations and some collaborating research institutes. These systems would serve basically two main purposes:

a) performing a kind of "broadcasting" function, information about IIASA would flow from IIASA to the "outside world" by means of the PRESTEL data base system

b) "interactive message exchange" would allow a certain kind of exchange of messages between IIASA and outside users.

The latter purpose would be served by PRESTEL's "replay-page" option [11] which permits the members of IIASA's closed user groups to have full exchange of text information with IIASA, through the use of alphanumerical keyboarding. One

limiting factor is that messages between two members of a closed user group cannot be exchanged using PRESTEL's present version. Special "short" messages, such as those containing names, addresses and telephone numbers, can be sent by any user of PRESTEL using any sort of keyboard; this function would be utilized, for example, when handling on-line orders for IIASA publications.

4. SUMMARY

As pointed out earlier, computerized text communication in an international environment is even now viable and useful. It can only be hoped that, when the present more or less "experimental" service becomes standard (like telex, telephone and mail) and available on future, large telecommunication networks, its quality will not suffer, and may even improve.

REFERENCES

[1] Turoff, M. and Johnson-Lenz, T. & P., How to use EIES Electronic Information Exchange System - Research Report No. 7, Computer Conferencing and Communications Center at NJIT (New Jersey Institute of Technology, 1977).

[2] The Planet and Notepad Systems - User's Guides (Infomedia Corporation, Palo Alto, 1979).

[3] Computerkonferenzsystem KOMEX, Version 1, Benutzungsanleitung, Institut fuer Planungs- und Entscheidungssysteme (IPES)(Gesellschaft fuer Mathematik und Datenverarbeitung, Bonn, 1980).

[4] IIASA '79 - Annual Report (International Institute for Applied Systems Analysis, Laxenburg, Austria, 1980).

[5] Vallee, J., Lipinski, H.M. and Miller, R.H., Group Communication through Computers, Volume 1: Design and Use of the FORUM System, Institute for the Future Report No. R-32 (Institute for the Future, Menlo Park, California, 1974).

[6] Pearson, M.L. and Lathrop, C.L., Using the Computer to Communicate: A User's Guide to TELECENTER, IIASA Working Paper No. WP-80-109 (International Institute for Applied Systems Analysis, Laxenburg, Austria, 1980).

[7] Gruebler, A. and Sheldon, R., Preparation of the IIASA-RSI Conference through Teleconferencing, Internal IIASA Paper (International Institute for Applied Systems Analysis, Laxenburg, Austria, 1979).

[8] Labadi, A., IIASA Gateway System and Experiments in Daily Operation, Proceedings of the COMNET '81 Conference, Budapest, May 1981 (North-Holland, Amsterdam, forthcoming).

[9] Smith, D.E., TELEMAIL Reference Manual, Volume I , the Basics (GTE Telenet, Vienna, Virginia, 1980).

[10] Ebenberger, H., Bildschirmtext, Elektrotechnik und Maschinenbau - Heft 12 (Springer Verlag, Vienna, New York, 1980).

[11] Benutzungsanleitung fuer Bildschirmtext-Teilnehmer (Entwurf)(Austrian PTT, Fernmeldetechnisches Zentralamt, 1981).

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
IFIP, 1981

The VMSHARE Computer Conferencing Facility

Charles Daney

Manager, Tymcom/370 Systems
Tymshare, Inc.
Cupertino, California
U. S. A.

VMSHARE is a computer conferencing facility developed in 1976 for use within Share, Inc., an IBM users' group. It originated from the need to solve problems of continuity, planning, and communication between meetings. This paper discusses its uses, structure, capabilities, implementation, and possible future development.

> ... we spend our time sending messages to each other, talking and trying to listen at the same time, exchanging information. This seems to be our most urgent biological function; it is what we do with our lives.
>
> [Lewis Thomas, "Computers", in The Lives of a Cell]

Introduction.

Share, Inc., an association of users of large IBM computer systems, was founded 25 years ago, in 1955. It seems generally agreed that the two major purposes of Share are the alleviation of redundant effort by the exchange of information and software, and the influencing of policies of IBM toward ends perceived as being in the common interest of both IBM and its customers. Over the years these purposes have been elaborated into a number of different themes. These include professional development, technical education, forecasting of trends in computing, and the like.

Currently, general meetings are held four times a year. Two of these meetings are known as "majors", and the typical registration is over 3000. The other two meetings are known as "interims" and have a much smaller attendance. Activities at both kinds of meetings include the following:

- announcements by IBM of new products
- tutorial presentations
- technical sessions
- working committees
- informal discussion sessions
- Share management meetings
- dialogs with IBM representatives on policies and problems
- drafting of requirements for product enhancements

In the early years of Share the relatively small size of the organization and the infancy of the computing industry permitted many technical innovations to come directly from the user group. These include the "Share operating system", one of the first assembly languages, and PL/I. In recent years, the major technical activities of Share have been conducted by a number of "task forces" chartered to research and report on "strategic" issues in computing such as large system requirements for application development, interactive systems, language futures, and distributed data processing. These task forces often meet with much more frequency than Share general meetings, in spite of the great expense of transcontinental travel.

The organizational structure of Share consists of four divisions. Each division is subdivided into "groups", "projects", and committees. The task forces are by definition ad hoc, are sponsored by one or more divisions, and are considered to be of temporary duration. Typically a group or a project is concerned with one particular component of the IBM product line, such as an operating system, a language, or an application like text processing.

In particular, one of the "groups" is chartered to deal with affairs related to the "VM/370" operating system. The activities typical of Share in general take a number of specific forms within the "VM Group". Because the VM/370 operating system has undergone rapid development within the last 10 years there has consistently been much discussion of new user requirements. A great deal of time is also devoted to critique of product changes made by IBM. It is also nesessary to discuss many software maintenance problems (bugs).

It is in this context that a computer conferencing facility for the use of the VM Group was conceived and implemented in 1976 by David N. Smith of Tymshare, Inc. The facility is called VMSHARE. In this paper the term VMSHARE will usually refer to the software that implements the conference, as distinct from the actual conference itself. The purpose of the remainder of this paper is to describe the uses, structure, capabilites, implementation, and possible future development of VMSHARE.

Uses.

Three of the major problems of Share are continuity, planning, and communication between meetings. These problems stem, of course, partly from the relative infrequency of meetings, which is in turn a simple result of the high cost of travel. But they are further exacerbated because Share is a volunteer organization which depends on contributed labor and resources to accomplish its ends. Continuity is a major problem because few individuals are able to attend all four yearly meetings, and a large percentage are unable to attend even both of the major meetings. This lack of continuity and the need to rely on volunteers clearly makes planning difficult.

So VMSHARE had its origins in an attempt to relieve the continuity and planning problems, as well as providing a medium for communication between meetings. Continuity of participation in ongoing activities is directly improved, because an individual can reasonably expect to contribute over a period of time even if travel is prohibitively expensive. Share general meetings are ordinarily very hectic events, particularly for officers, and so little planning gets done, even with the best of intentions. The

availability of a conferencing facility with electronic mail allows planning to be done in the comparative leisure between meetings.

However, by far the largest benefit of VMSHARE is the ability to communicate between meetings. In general the type of communication that occurs is much the same as at regular meetings. Given the principal interest of Share, this tends to consist largely of nuts-and-bolts discussion of IBM software and hardware. As is true at meetings as well, much of the participation is passive; for many, the greatest value of the system is the opportunity to listen in on colloquies of the "experts". Since contributions remain in the system until explicitly deleted, a very considerable body of lore has accumulated in the database, which is of great utility to both novice and expert.

Another class of mostly passive observers consists of representatives of IBM itself. Frequently the first notice of software bugs will appear on VMSHARE instead of through normal problem-reporting channels. Quite often, user-contributed software fixes will also appear (see Figure 2 at the end of the paper). Beyond that, VMSHARE provides IBM with useful market research information in the form of comments and general opinion on a wide range of software and hardware products. Users of VMSHARE are aware of this communication channel to IBM and make frequent use of it to organize support for their particular current enthusiasm. IBM is supplied monthly with a copy of the database contents, and this data is made widely available internally on IBM's own network.

Several other VMSHARE-like conferences have been established for other groups within Share, including several of the task forces. These other logins use the same software but different data bases. As of this writing, two additional conferences have also been set up for ANSI standards committees.

VMSHARE as a conferencing facility is in every respect a production system. It is available for use essentially 24 hours a day, every day. Over 300 participants are validated, from all parts of North America and Western Europe. There are more than 1100 sessions a month. About 800 different MEMO and PROB files are currently in the database.

Structure.

VMSHARE is different from many other computer conferencing facilities in one respect. It does not attempt to emulate closely the style and procedures of traditional face-to-face conferences. There are no intentional close analogs of subconferences, committees, sessions, or the like. Instead, everything is oriented around a collection of files of various types. Since a given file will generally deal with a single fairly narrowly defined topic, it is possible to view the file as the analog of a conference session, but it is hardly necessary to insist upon this identification. Likewise, since access to specific files can be restricted to any given subset of conference participants, it is possible to simulate a closed committee. But again it is not necessary to look at it that way. Rather than the emulation of any set of particular social communication structures, one of the guiding principles in the development of VMSHARE was simplicity of both concept and implementation.

Each file in the system has an arbitrary (up to 8-character) name and exactly one of a small list of specific types. Presumably the name is of mnemonic significance. However, if it is desired to establish some more formal naming conventions, that is entirely possible.

So files are the basic objects in the system. There are then defined a number of commands which operate on these objects. Most of the commands will be mentioned later, but it can be observed that they fall into a few basic categories: display of files in various ways, creation of files, modification of files, movement of files, and identificaton of particular subsets of files. There are miscellaneous other commands for control of the interactive session, requesting help, etc.

The "type" of the file is one of the following: MEMO, PROB, REPORT, NOTE, LOC, SCRIPT, LISTING, PROC, ADDRESS, HELP, RES, or MAIL. A file's type determines a number of things about it: the scope of the name, (default) physical characteristics, and what commands can operate on it. By convention, the type also gives a hint about the kind of information contained in the file, but the conventions can often be ignored if desired. Other file types could be defined easily, and almost certainly would be in other contexts.

The two most common types of files, the files that form the basic content of the conference, are MEMOs and PROBs. MEMOs are files which contain ideas, information, or questions. They may be created by anyone and seen by anyone else (unless restricted by an access list). PROB files are just like MEMOs with the single exception that, by convention, their contents are related to problems with hardware or software or support. A NOTE file is just like a MEMO file except that its existence is not indicated by commands that report new files. A NOTE is, however, still a public file unless it is protected by an access list. NOTEs exist for the sole special purpose of allowing "background" discussion which need not "clutter" up the main conference.

A "discussion" on VMSHARE proceeds as people use the capability to append new information or comments to a MEMO, PROB, or NOTE file. A MEMO or PROB file generally represents a discussion with a rather narrow focus. More broadly focused discussions are represented simply by collections of related files. Any conference participant can initiate a new topic or train of thought at any time. This is not so confusing as it may sound, because any other participant may choose to ignore any colloquy that does not interest him.

In Figures 1 and 2 (at the end of the paper) are included two short but representative files, to give some of the flavor of VMSHARE. Figure 1 exhibits a typical MEMO file, which in this case is a request for information. Notice that VMSHARE has automatically included lines which tell when and by whom the file was created and appended. In this case, the question received an answer in less than two days.

In Figure 2 we have a PROB file in which someone reports a programming error he has found in the control program (of VM/370). The problem is not only reported, but a fix is offered as well. PROB files thus provide a means of distributing software corrections very rapidly. However, a PROB file may merely state a problem or enquire whether anyone has ever encountered a problem in a given piece of software.

MEMO files may be used for much besides the asking of questions. An individual may volunteer information or state opinions. Actual code can be placed in MEMOs, providing a program distribution mechanism, at least for small pieces of work.

When a discussion has reached a certain level of maturity, it may be crystallized in a more formal file called a REPORT. These are often (but by convention only) written by an individual or a committee, expressing the perceived consensus of many discussions. An important difference with a REPORT is that it can only be modified by its owner (i. e. author); appends are not allowed.

All of the file types so far discussed are considered to have a "global" scope. This means that they are visible to all participants, unless protected by an access list. The individual who creates the file becomes the owner. If an access list is also created, only the individuals in the list are permitted to see, or even know of the existence, of the file. Access lists can be created or changed at any time by the file owner. Except for REPORTs, a global file can have information appended to it by anyone permitted to see it. Only the owner, however, is able to edit and change the file arbitrarily.

Another class of file types is private and "local" in scope. These local file types are provided for various special purposes. Local files with the same name but belonging to different individuals are completely distinct. A LOC file is a completely general purpose local file. LOC files are often used as a notebook or scratchpad for developing initial drafts of memos which will eventually be made public. Alternatively, they may contain memoranda to oneself, copies of mail received, etc.

VMSHARE allows command procedures, which are stored in local files called PROCs. A PROC is simply a sequence of commands which are stored for repetitive execution. They also allow for the substitution of variable symbols, to permit the creation of generalized procedures. The most important PROC is the PROFILE, which is invoked whenever the user who owns it logs in. This is used to establish a preferred, customized environment.

A third type of local file is the SCRIPT file. This consists of text interspersed with formatting commands for processing by the "Script" text formatter. The output of the Script formatter is placed in a file of type LISTING.

The remaining file types are "special" in some respect or another. MAIL files are in this category. An ADDRESS file is kept for each user, but its use has nothing to do with MAIL. It simply contains the address of each user for the purposes of mailing printer listings and for the information of other participants. HELP files contain information and instructions related to use of the system. RES files, finally, are for use by the "resolutions" secretary. This latter individual has the SHARE-specific function of maintaining the group's "resolutions", i. e. statements of requirements addressed to IBM.

Commands.

Files are, most often, created by the EDIT command. An EDIT command specifies a file type and name. The command invokes the "standard"

editor of the interactive system under which VMSHARE is running (in this case CMS). If the file does not exist the user simply enters text. If the file already exists the user may alter its contents, provided he is the owner. If he is not the owner, he may still employ the editor to browse through the file, but he is prevented from changing it.

There is an append-only update mode for MEMO, PROB, and NOTE files. Anyone to whom the file is visible may issue the APPEND command for it. This causes him to be placed in the editor, where he composes his append. The contents of an already existing file may also be appended to another file.

There are several file movement commands: COPY, TRANSFER, READ, and RECEIVE. COPY is straightforward and obvious. TRANSFER has two similar functions. Within VMSHARE TRANSFER can be used to change ownership of a file. In this case no actual data movement takes place. TRANSFER can also be used to "export" files to another conference simply by naming the other conference. The companion to TRANSFER in this case is READ, which "imports" a file transferred from another conference. RECEIVE applies only to MAIL files and has the capability of placing selected mail in a file.

To round out the collection of file utility commands, there are DELETE and RENAME, both with the obvious function.

A number of commands exist for displaying file contents. The most straightforward of these are TYPE and PRINT. TYPE shows file contents on the user's terminal while PRINT sends them to the high-speed system printer. A third rather specialized command of this sort is NETPRINT which is intended to drive a network-connected printer at up to 4800 baud. All three of these commands can be specified for many files at once. The files to be processed can be selected on the basis of name, type, or date of last change (or all three).

All three commands also have a special mode called "brief" which shows, for files which can be appended, the appends added since a specified date. Thus TYPE BRIEF NEW is a single command which can be issued to display on the user's terminal only the appends to MEMO and PROB files added since the user's last logout, and all new or changed REPORT and HELP files as well. During the display of any file, typing may be interrupted and will resume with the next file, if any.

Since the high-speed system printer is likely to be located at quite a large distance from the user, all printed output must be mailed. The address is determined from the ADDRESS file. There can be multiple ADDRESS files - a default and others which can be selected by the SET MAILADDR command.

All of the preceding file display commands show the file in chronological order. For those files which can be appended, the REVIEW command does just the opposite, displaying appends in reverse chronological order. The user may interrupt this display at any time when he has seen enough.

The EDIT command may be used, as previously noted, to browse through files. For MAIL files there is also the RECEIVE command, which by default displays mail on the user's terminal.

"Information overload" is a recognized problem with computer conferencing facilities. To address the problem, VMSHARE provides a number of commands which select subsets of existing files according to a variety of criteria. These file selection commands list file names and various other kinds of information on the user's terminal.

One selection criterion is date (and time) of last file modification. The simplest command that uses this is QUERY SINCE, which takes a parameter of date (and optionally time). The names, types, dates of change, and numbers of lines of MEMO, PROB, HELP, and REPORT files modified since the specifid date are displayed. A variation of this is to use NEW instead of SINCE; the date and time used is that of the user's last logout from the system. QUERY NEW is done automatically for the user at each login.

Somewhat more powerful is the LIST command. In addition to allowing date and time specifications via NEW and SINCE options, LIST permits file selection on the basis of type and name. A specific type may be designated, or "*", meaning all types. A specific name may be disignated or (more likely) a "name pattern". In a name pattern, the character "?" is a wildcard character which matches any single character. The character "*" stands for any string of 0 or more characters in a name pattern. A logical not sign ("\") preceding any character except "*", "?", or "\" will cause a match with anything except that character. (The characters "*", "?", and "\" are not legal in file names.) Thus, for example, "\A*" matches any name not starting with "A". Name patterns are accepted also by a few other commands like TYPE and PRINT.

One other selection criterion available is LOCAL or the user-identifier ("userid") of some other conference participant. Selection criteria used together have the effect of taking the intersection of selected sets of names. The names of all files selected by the LIST command are displayed. If the file is of type MEMO, NOTE, PROB, REPORT, or HELP, the first line of the file is also displayed. By convention, the first lines of such files contain a descriptive title.

Similar to LIST is the FILES command. The same selection criteria are available, but the output is a summary of file characteristics (such as number of lines, date of last modification) instead of the first line.

The final file selection command, and a very powerful one, is the INDEX command. The INDEX command gives the user access to an on-line index of the contents of MEMO, PROB, and NOTE files. The index is built by taking every word in each file, translating to upper case, and truncating to 16 characters if necessary. The index is updated in real time whenever someone updates a file. There are several hundred common words excluded from the index. After entering the INDEX command, the user is prompted for a keyword. The list of files which contain that word (if any) will be displayed. "*" is a wildcard character when used at the end of a key - in that case all keywords having the same stem will be displayed. The INDEX command is terminated by entering a null line. This index facility is expensive in terms of space required to store the index but is worth the expense.

There are two commands provided specifically for operating on MAIL files, although most of the others (including EDIT, TYPE, PRINT, DELETE, and LIST) also apply. Taken together all the available

commands provide a decent electronic mail capability. Mail is created with the SEND command. The message to be sent can be entered on the same line as the command. The message can also be contained in some already existing file (usually a LOC file). If the mail is not specified in one of these ways the user is placed in the system editor to compose his message. After the SEND is complete, the text of the message is contained in a LOC file with a name of LASTMAIL. This permits the same mail to be sent to selected other users. Mail can be sent to many users simultaneously by including the wildcard character "*" in the userid designated as recipient. Thus a userid of "X*" causes the mail to be sent to all users whose identifier begins with "X". This is useful, of course, only if there exist conventions for the naming of userids.

Incoming mail is displayed with the RECEIVE command, which is a variant of TYPE. RECEIVE will prompt the user after all mail has been displayed as to whether the mail should be deleted. RECEIVE has the additional function of placing incoming mail in a file, if desired. Any mail that has not been deleted, either from or to a user, may be identified with the LIST command.

One particularly nice feature available is "mail echo". If the intended recipient of a message has a LOC MAILECHO file, it will be displayed to the sender before he composes his message. This allows one to leave a notice in the system if one expects not to be using it for a while or to recommend that mail be directed to another userid.

VMSHARE has access to a text formatting program called Script, which exists on many IBM systems but in many variants. The text formatter takes a file which contains imbedded formatting commands and produces camera-ready output on a system printer. Output can also be directed to a file. The program is invoked with the SCRIPT command. The input file must have a type of SCRIPT. If output is placed in a file, it will be of type LISTING.

Because the database of files in a VMSHARE-like conferencing facility may grow to be quite large, a great deal of time (to say nothing of paper) may be required to PRINT even a small fraction of them. For this reason, the DEFER command has been provided. The arguments to DEFER consist of another command, followed by its appropriate parameters. At present this other command may be only LIST or PRINT. The effect is that the command is queued for later execution in a batch mode, thus saving time during the terminal session. LIST output in this mode goes to the system printer. The status of "batch jobs" queued for a user can be determined with the QUERY DEFER command. Deferred commands can be rescinded with the CANCEL command.

Online user assistance is available with the HELP and HOW commands. As already noted, there are special files in the system of type HELP. These are created only by the system administrator or someone he designates. The HELP command produces a summary of what is available, as does LIST HELP. HELP files can be displayed with the HELP command, and also by TYPE and PRINT.

The HOW command is used to display the syntax and general description of any available command. The information it displays is extracted directly from the general system reference document.

The SET command is used for various kinds of session control. One

useful instance, if the user has an ASCII CRT, is SET CRTSIZE. If a number is set for the CRT size (number of lines on the screen), then most commands which do output will pause for 30 seconds every time that many lines have been sent. There is also a TERMINAL command to define other characteristics of the user's terminal, such as line length, half or full duplex, etc.

The QUERY command is used to ascertain the values established by SET and TERMINAL. It may also be used, as already noted, in the form QUERY NEW/SINCE to list changed files or QUERY DEFER to list batched commands. QUERY LASTUSE is used to determine the last time a giver user logged in to VMSHARE. QUERY NAMES identifies other users who may be logged in at the same time. Simultaneously logged in users can communicate with the MSG command.

Two commands are related to security. With the PASSWORD command, any user can change his login password. With the ACCLIST command a file owner may create or delete an access list for his file. The access list is simply a list of userids which are permitted to know of a given file. The file is invisible to anyone not in the list. A userid name in the access list may contain the character "*", as with mail, to designate a group of users.

VMSHARE permits a simple sort of command procedure called a PROC. A PROC, as already noted, is a specific type of file which contains VMSHARE commands and possibly some special PROC control statements. Invocation of commands in command mode finds PROCs ahead of normal system commands. Invocation of commands inside of PROCs finds PROCs ahead of system commands unless the command has the same name as the current PROC, in which case the system command is called directly. At login time a PROC named PROFILE will be invoked automatically if it exists.

PROCs do not have facilities for looping or user defined variables. They do, however, have system defined variables and control statements like IF and GOTO.

Implementation.

VMSHARE runs under the CMS component of the IBM VM/370 operating system. It is written almost entirely in the Exec language, which is the standard command macro language of CMS. Because Exec is a command language, much of the actual work can be done by quite standard CMS commands. A very few special-purpose functions have been written in 370 Assembler language and PL/I, but these are still invoked from Execs.

Since Exec is an interpreted language, the performance of VMSHARE is not what could be achieved by a system written in a compiled language, but it is usually satisfactory. In fact, the PROC processor is itself written in Exec. Thus we have an interpreter interpreting an interpreter. Performance of PROCs is therefore not exceptional. VMSHARE was written in Exec because of the power and flexibility of the language. VMSHARE began as something of an experiment, and most steps in its development were made in the spirit of "try it and see if it works". The Exec language was very well suited for this kind of development because of the great ease of making and testing changes.

Each VMSHARE session runs in a single "virtual machine" in a large

multi-access Tymshare VM/370 time-sharing system which is running many other applications as well. The software supports any number of simultaneous sessions. Public files are fully shared, but appropriate interlocks are maintained to prevent simultaneous alteration of the same file. This is accomplished by a file server program running in a separate virtual machine. A non-shared single-user mode of operation is also possible. VMSHARE, or any of the other conferences, may be accessed through the Tymnet international public data network. A majority of users are able to do this from their own places of work, or even homes, with a local phone call.

So VMSHARE is basically just another time-shared application system. One of its interesting characteristics, however, is that it is a "limited" or "secure" environment. That is, special care is taken to ensure that a user is unable to leave the environment. From login to logout, everything the user does is under the control of the VMSHARE software. The user has no access to the general capabilities of the time sharing system. This is done because access to those capabilities would permit easy compromise of privacy and integrity of data files, and also to limit usage to the intended purpose. While this "padded-cell" environment may seem like an obvious requirement, its implementation necessitated certain changes to the operating system. As it turned out, this was much easier to do than to write a new interactive system to replace CMS. The software can still take advantage of all the facilities of CMS, but in a controlled way.

For each individual conference, one participant is designated as the system administrator. The role could just as well be played by several people, because as far as the software is concerned, the administrator has no special privileges. Mainly his role is a social one: to handle complaints or requests to validate new participants in the conference. Whether he actually assumes a special leadership or coordinating role in the actual work of the conference is not predefined. There are no specific facilities available to the administrator, or any other individual, to control what information is introduced in the conference. The administrator, or other participant, establishes his right to perform such functions as validating new participants by first issuing a special undocumented command which grants such privilege if a password, determined when the conference is initialized, is correctly entered.

Future directions.

Four years of experience with VMSHARE have taught many lessons about computer conferencing. Many improvements have been incorporated as a result of this experience. In this section we focus on indicated directions for future development.

1. Techniques for searching the database.

A relatively simple extension of the INDEX command would permit the construction of result lists and boolean operations on those lists. The lists could then be used to drive other commands such as TYPE and PRINT. In this way complex qualified searches could be performed. At the same time, it is desirable to provide access to a general, sophisticated information retrieval program. This would permit incorporation of true indexed databases into the conferencing facility. The conference thus becomes a specialized information

utility. This is something which is badly needed in many conferences. For example, in VMSHARE it would be highly desirable to have an easily searchable database of requirement resolutions, or of Share member characteristics, or of software available for exchange. Another database application would be bibliographies and other resource catalogs. The bibliographic data could allow for short annotations and comments, permitting active involvement of the users.

2. The file system.

The VMSHARE approach of providing a general purpose file system for computer conferencing, as opposed to a specialized, highly structured system, is judged to be successful. One departure which has been considered, however, is to replace the file as such with a new building block called a "segment". The segment becomes the smallest identifiable unit of textual data; a file is an ordered collection of segments. The one big advantage this offers is that ownership of the segment remains with its author. In the current system, once data has been appended to a file, only the owner of the file can alter the append. A segment could be updated or deleted at any time by its owner. A single segment could be part of any number of "files", and only a single copy of the text need be stored. Complex network-structured "hypertext" files could be constructed out of segments.

It would be nice to keep additional data in the file directory. At present, the directory records little more than the size (length and width) of a file, who created it, what its name and type are, and when it was last updated. In addition, it would be useful to know when the file was created, whether its contents are indexed, when it was last read, etc. The directory could also contain a few keywords for the file and a short title.

The need for a more elaborate access control mechanism is also sometimes indicated. In addition to access lists, it should be possible to refer to specific groups and the type of access permitted (read-only, append-only, or full-update). The need for this is most urgently felt in order to support joint authorship of reports and papers. This should mesh smoothly with any faciities provided for defining (sub-)groups within the conference. At a minimum, groups should not be restricted to a simple hierarchical structure. But this really touches on the next area.

3. Intercommunication of multiple conferences.

Already the several conferences which have been defined have had a need to exchange mail or files among themselves. They also have overlapping memberships. Although it will always be possible to set up fully autonomous but "loosely-coupled" conferences, more facilities are needed for "tightly-coupled" conferences. For example, there should be a single directory of users and a single directory of files. Both directories would indicate with which conferences each entry is associated. This arrangement makes membership of individuals and files in multiple conferences quite easy. However, it may still be useful to maintain the concept of participation in only one conference at a time in order to control information overload, as long as context switching is simple and easy.

4. Bulk transmission of data files.

There are two related requirements in this area. On one hand, conference users feel a frequent need for hardcopy of conference data files. This is especially true if the user's terminal is a CRT, but in general good quality printed output is much more satisfactory for study, markup, and archival than is the output of most terminals. The second requirement is for transmission of data files in machine-readable form, for example programs or numerical data. If the conference user has access to another computer with printing facilities, the first requirement can be handled if the second one is. But in any case, it is still desirable for the conferencing facility to be able to communicate with high-speed RJE equipment, which is essentially just a fast terminal.

It appears that communications costs could very quickly dominate the economics of computer conferencing. Already it is possible to provide the hardware, software, and operational support for a computer conferencing facility capable of supporting 75 to 100 simultaneous users for a cost of about $20,000 per month. The communications costs (network tarrifs) that would have to be paid if such a system were fully utilized would be at least one and a half times that. And the cost of hardware is declining much faster that the cost of communications. Furthermore, the data bandwidth of current network facilities is severly limited in comparison to what could be used. All of this indicates that communications will be a limiting factor in computer conferencing; at least it will constrain the desirable growth in bulk transmission of data files.

There is another way to quantify the significance of communication costs. Currently, network connect time costs about $4.50/hour. The cost of host computer resources is much more variable, depending on what the user is doing. It reaches a high of $10/hour or more if the user is actively printing files, searching the index, etc. But it is much less when a user is creating new text, certainly less than the connect charges. These charges therefore become very significant in a system where user input begins to rival information retrieval in importance.

Hence an important trend is the increased use of personal computers for the preparation and formatting of text for input to computer conferences. There are important psychological benefits to giving the user the ability to create text on his own computer, in addition to the saving of network and telephone connect charges. Some processing load is also removed from the central facility, and if the conferencing facility is constrained in number of ports then it too will benefit from shorter connect times. But the same amount of data (almost) must still be transmitted to the conference, so total network bandwidth is still a problem. If the conference user has good printing equipment, he may also want to copy files from the conference in order to do his printing offline.

5. Interconnection of multiple hosts.

Currently, all VMSHARE-like conferences run on a single host. It is no great extrapolation to expect that in the future they will be on many hosts, and so communication among them becomes more challenging. Indeed, we would want facilities to allow communication with other quite different conferencing systems available on the network or via gateways to other networks. Such connections may be used for interactive sessions in a "pass-through" mode, or for access to specialized databases or other resources not available on the primary host, as well as for bulk data transfer.

But even when direct connection of hosts is not possible, if there are good interfaces for personal computers, then a personal computer with adequate local storage can act as an intermediary for bulk data transfer, by first logging into one system, copying data to its local storage, then logging into a second system and sending the data (manual "store and forward").

Just as people will become comfortable with preparing text files on their personal computers, they will be most comfortable with the tools and facilities available on the conferences or hosts where they do most of their work. Hence there is much to be said for a system which allows users to do most of their work on a preferred host, while having the ability to make brief contact with a chosen conference to exchange information.

All this of course raises a vast number of questions about the mechanisms for open system interconnection, file transfer protocols, and so forth. To take just one example, almost anyone who uses interactive computers now has access to at least one electronic mail system. Many have access to half a dozen or more different systems. It is becoming a major undertaking just to read and process the daily mail. So it is highly desirable to be able to receive all mail with a single command, which implies the need for a way to connect one mail system to another.

> All 3 billion of us are being connected by telephones, radios, television sets, airplanes, satellites, harangues on public-address systems, newspapers, magazines, leaflets dropped from great heights, words got in edgewise. We are becoming a grid, a circuitry around the earth. If we keep at it, we will become a computer to end all computers, capable of fusing all the thoughts of the world into a syncytium.
>
> [Lewis Thomas, "Computers", in The Lives of a Cell]

Figure 1.

```
* VSAM PERFORMANCE UNDER CMS
HAS ANYONE INVESTIGATED THE PERFORMANCE OF VSAM WITH CMS
OR THE LACK OF IT?
PLEASE APPEND ANY RELATED COMMENTS. THANKS FOR YOUR HELP.
ED SPONGBERG/EAF

*** CREATED 08/03/79 17:44:53 BY EAF ***

don williams, at amoco, has done some performance
comparisons and found that vsam under cms stinks.
you can call him on 918 664 3000 for details.

*** APPENDED 08/05/79 09:14:36 BY $PR ***
```

Figure 2.

```
Bug in DMKCCW doing SPACE COUNT CCW

If a space count CCW is issued from a virtual machine
with the length less then or equal to 3, the translated
CCW will have length data of zero!
VM release 5 plc 6, but it should fit all releases.
DMKCCW
./ R 1851000 $ 1851100 100
        bc    13,dasdxfjn

it only showed up during diagnostics testing of a CDC
disk drive.  it worked standalone, but gave incorrect
length errors virtually
john alvord

*** CREATED 01/10/79 01:18:27 BY AMD ***
```

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

CREATING AN ADAPTIVE COMPUTERIZED CONFERENCING SYSTEM ON UNIX

Michael M.L. Pearson
James E. Kulp

International Institute for
Applied Systems Analysis
Laxenburg, Austria

Telecenter, a computerized conferencing system, was implemented with approximately one man-week's programming effort by exploiting tools available on our institute's computer time-sharing system. Telecenter's adaptive design philosophy was shaped by user needs, and its rapid implementation was facilitated by a UNIX-based computing environment permitting flexible, modular design. Appendices contain a sample session using Telecenter and details of UNIX system features underlying its design.

INTRODUCTION

This paper discusses a computerized conferencing system being developed at our institute. It does not describe the system. A brief sample session is in Appendix I. For a more detailed description we refer the reader to Pearson and Lathrop (1980). The system, which we call Telecenter, is similar to others of its genre; using it one can create conferences, add participants, enter new conference comments, modify old comments, learn who has seen which comments in any given conference, get an overview of one's own status in all one's own conferences, and so forth.

Telecenter's current form, however, is not the topic of this paper. What we wish to address here is how and why Telecenter was developed. The "how" has to do with our institute's UNIX[1] operating system and is the subject of the last part of this paper. The "why" has to do with our users and their needs and is addressed in the first part. In our approach to both users and Telecenter we have tried to conform to certain general principles recognized by others involved in designing and implementing computerized conferencing systems, office automation and decision support systems. One result, we believe, is that we have made considerable strides with surprisingly few resources. For example, Telecenter itself was created with approximately one man-week's programming effort. This is noteworthy considering the usual requirements for developing such a system. Murray Turoff, a pioneer in the field of computerized conferencing writes in The Network Nation:

> At the moment, most of the (computerized conferencing) systems that have been implemented represent many person-years of programming effort. There is no suitable higher-level language that allows a concise and appropriate specification of a human communication process. All current efforts at using FORTRAN, APL, BASIC, etc. are like describing a picture of the Mona Lisa in words only... Currently we are beginning to understand the basic functions that characterize conferencing systems and it can be expected that a computer language will evolve that can allow these systems to be created in a few person-months of effort (Hiltz and Turoff (1978) p. 391).

The kind of higher-level language for human communication processes that Turoff speculates about has yet to be invented.[2] Relevant to this statement, however, is the fact that software tools used to construct conferencing systems a few years ago are inferior to tools developed more recently. That Telecenter was made operational so quickly is primarily the result of its having been fashioned under an interactive and comparatively state-of-the-art time-sharing operating system, but it is also partly the result of our attitude toward user needs.

USERS' NEEDS: THE "WHY" OF TELECENTER

The institute where the authors are employed, the International Institute for Applied Systems Analysis (IIASA) is an interdisciplinary, nongovernmental research institute, chartered in October, 1972 upon the initiative of the academies of science or equivalent institutions of twelve nations (there are now seventeen National Member Organizations). By applying systems analysis, its staff of scientists from East and West is seeking to gain a better understanding of important contemporary problems resulting from scientific and technological development. IIASA conducts much of its research in cooperation with other research and policymaking organizations worldwide.

One example of a IIASA research activity seeking better international communications tools is the Survey Project (soon to become the task on the "Craft of Applied Systems Analysis.") The Survey Project has as one of its responsibilities a series of books on applied systems analysis and, therefore, the job of coordinating and supporting the work of a world-wide group of author/specialists. Since one of the authors of this paper works at the Survey Project--primarily a publishing activity--his first involvement with computerized conferencing was in the specialized form he terms "computerized manuscript conferencing," the use of computer conferencing mechanisms to facilitate the joint authorship of manuscripts (see Pearson, 1980). As an outgrowth of this his interests have broadened to include any application of computerized-conferencing-type tools to facilitate applied systems analysis and institute research, what he, for want of a better term, calls "using the computer to communicate" (see Pearson, 1980). Manuscript conferencing remains central to the project's teleconferencing interests since a large percentage of the institute's research involves writing, but it is now a part of a more general activity whose goal is to explore the possibilities of a new kind of computing/communications environment--one

which should be able to significantly expedite scientific work among institute staff separated in time and/or space, and at the same time be accessible to the average researcher.

It is our belief that the best way to explore "using the computer to communicate" for scientific research at our institute is to facilitate it for the applications of individual scientists. Thus as a tool-environment, Telecenter offers a set of potentials that we do not fully understand and will not fully understand until we involve users and their own potentials and see how matters evolve. It is these users, the IIASA scientists and their applications, that we must serve in order to make this medium serve applied systems analysis.

An example of how Telecenter is useful in facilitating institute research is a recently initiated IIASA study centered on atmospheric change and involving, in its initial stages, a group at IIASA (Austria), at the U.S. National Academy of Sciences (Washington, D.C.) and a consultant in California. The key person at IIASA, who had never before used a computer system for word-processing or any other "non-scientific" application, was quickly introduced to both the standard UNIX editing/text-processing utilities and Telecenter (Telecenter incorporates the standard UNIX editing programs, see infra.) The individual in California, already familiar with UNIX facilities, easily accessed Telecenter via IIASA's host facility on TYMNET. As special needs became evident--a manuscript conference for organizing atmospheric impact descriptors for a game they are developing--it was possible to offer specialized Telecenter functions for organizing this information. Whenever hard copy is needed, full text-formatting facilities are available that interconnect with Telecenter via the common time-sharing environment.

Similarly, Telectr has been used to facilitate an exploratory effort at IIASA to produce an electronically collected Decision Support Systems Newsletter. Telecenter has both facilitated communication among individuals involved in early discussions of the concept, and, in a modified form, provided a computer environment for organizing submissions for the newsletter's first mock-up. Throughout its use, Telecenter has had to adapt to specialized needs of IIASA researchers. If a new function is requested, it is usually available in hours and without recourse to professional programmers.

The key to Telecenter is its flexibility. Under no circumstances do we want to create an elaborate piece of machinery on the basis of our, undoubtedly premature, perceptions as to what is best for institute staff. Hence the need for modules, adaptable functions, that we can offer as preliminary tools to individuals for whom it is not fully clear at the outset in which direction their activity will take them.

We agree entirely with what Uhlig, Farber and Bair say regarding the implementation of computer systems for the "Office of the Future" and think it especially applicable to systems such as Telecenter:

> Too often systems are built to be used by and for their designers--not by the ultimate users. We can see this in many current systems. Also we tend to assume that the world will stop and wait for our systems to be

> fixed and that the users will be tolerant of loss of information. Many of the systems of the future will be used not by computer people but by executives, by the public, and by policy makers (Uhlig, et al., (1979) p. 153).

It should be stressed that we are by no means certain that a computerized conferencing system offers the best solutions to the problems of facilitating joint research electronically. Our primary objective is facilitating research, not creating a conferencing system. It may be a mistake, for example, to try and enforce the structured notion of conferences--topic oriented discussions--on the normally fluid communications habits of most researchers. Or it may be futile only in some cases, not in others, depending on the individuals involved and the task at hand. It is most likely that we need an environment that integrates conferencing, computer-mail functions and other computer-based tools--text processing, database use, computational routines, and so forth. The important thing is to have a facility up and running that can be adapted to user needs.

Thus, Telecenter is not primarily the kernel of a teleconferencing system, but, rather, a set of modifiable functions that can serve as a starting point in searching for a suitable environment for "using the computer to communicate". It is important that these functions can be packaged in a single, consistent, easy-to-use system such as Telecenter but even more important that they exist as modules that can be modified and combined in different ways for different purposes. Of all the computer-based, person-to-person communications tools we are familiar with, computer conferencing seems the most accessible to people unfamiliar with computers. It is for this reason we began with a general computer conferencing schema.

Service as another word for availability

Our point of departure in exploring the potential of "using the computer to communicate" at our institute has been the user. Similar kinds of attitudes have emerged from decision support system (DSS) research. Consider Gambino and Keen's admonitions to learn the language of the group you are supporting, and develop command functions for it, "build a capability rather than a product"; support first and extend later (as inexpensively as possible); select innovative, creative users with something at stake; support the person rather than solve the problem or build a model; encourage feedback and respond to user ideas and requests (see Gambino (1980)).

And it is our experience that, providing the user is motivated and has an important job to do, the single most important factor influencing the success or failure of an activity is availability. This concept runs as a common thread through all phases of implementing the computer/communications environment mentioned above--from the simple existence of Telecenter (before it we had nothing we could use locally) to documentation of facilities, guaranteeing acceptable response time (the single biggest problem to date in promoting use of the system; other imperfections in the system have been insignificant factors compared to this) and access to terminals.

A corollary to availability is adaptability. If a user needs a certain feature available in order to do the job at hand better, Telecenter must provide it--adapt to the user's need. This fact, even more than the issue of cost, precluded our turning to already existing conferencing systems. Most were not portable: they were written in impossible-to-modify assembler language or in large, unwieldy programs in higher-level programming languages that were unsatisfactory. Some constituted whole systems--worlds unto themselves--and none was suitable for integration into our institute's computing environment. We needed something we could quickly implement, understand and adapt.

An individual will only use a computer system such as Telecenter when it is preferable to competing tools such as pencil, telephone, letters, telex, assistants, and so on. In DSS terminology, the system is discretionary--it is at the discretion of the user/decision maker whether or not to employ it (see Sprague (1980)).

We can learn a number of lessons from DSS designers and implementors. Telecenter-type functions could, we think, evolve into a communications tool with significant DSS components, although at present it is not a DSS; its primary purpose is not offering models and databases to help managers make decisions. Nevertheless, Telecenter's relation to its users is analogous to that of a DSS to its users. The DSS notion of adaptive design, for example, is applicable to our approach. As Sprague says, it involves:

(1) Identifying a critical subproblem

(2) Building an initial system

(3) Using and evaluating

(4) Adding and deleting capabilities

(5) Repeating 3 & 4

Similarly, as far as Telecenter's development is concerned, the concept of a DSS generator, and that of DSS tools, both have close parallels with, respectively, the script mechanism of the UNIX command interpreter (see below) and the UNIX operating system itself. In DSS terminology, the DSS generator is the vehicle that permits the builder of a specific DSS do his job quickly and efficiently without having to have the computer expertise of a systems programmer. Thanks to the DSS generator, the builder can devote himself exclusively to decision tasks at hand--analyzing applications issues and responding to them. "DSS tools" are the products of the systems programmer (the "toolsmith") that permit the existence of the DSS-generator facility.

An important issue here is willingness to trade off responsiveness to user needs against polish in final design. Instead of dedicating all resources to the creation of a perfect, fully designed system, the approach is to devote, say, fifty percent of one's resources to initial system design and implementation then devote the remaining fifty percent to a labor intensive service activity that modifies the system while simultaneously working

with end users. Thus we have found that a large component of research into the usefulness of this new medium is service--the promotion of availability in its broadest sense. Implementors must have as their primary perspective, and responsibility, both facilitating and experimenting with the use of such tools to assist researchers. The luxury of a flexible and adaptable system does not spare one the expense of human time spent directly assisting users during many sessions.

ADAPTIVE TOOLS: THE "HOW" OF TELECENTER

After pointing out the significance of declining costs of hardware in the microelectronics field for forthcoming developments in "Office of the Future" technology, Uhlig, Farber and Bair go on to say:

> However in order to effectively utilize all this cheap logic we must reduce the cost of the software. We must create simpler systems that can be built reliably with small software investments. Some of the things that contribute to declining costs are utilizing single-task-machine approaches. This vastly simplifies the cost of software. In addition having simple interfaces simplifies the overall task. Furthermore, the adoption of simpler program structures will vastly ease the task of the programmer. The use of highly modular code may increase the use of cycles (which are cheap) but will greatly reduce the cost of software complexity. The use of structured coding technology will also contribute to the reduction of the cost of programming. But perhaps the greatest contribution to reducing software costs in the Office of the Future will be to create a better software development environment, i.e., more and better tools presented in a fashion that makes them always available to the programmers. (Uhlig, et al., (1979) p. 173)

As far as the creation of Telecenter is concerned, we believe that UNIX offered one such "better software development environment" and that four features of our UNIX system deserve mention in this regard:

(1) its utility programs and the philosophy behind them,

(2) its command interpreter,

(3) the transparency it offers between functions implemented at different levels of software, and

(4) its hierarchical file system.

Although he is focussing primarily on tools for making compilers (for example, a lexical analyzer and a parser) Stephen C. Johnson in Language Development Tools on the UNIX System makes points about the philosophy of operating system software tools relevant to the building of Telecenter. We feel these points are of interest here, first, because Telecenter consists largely of UNIX utilities and, second, because it has a highly modular design.

> Over the years, a number of general utility programs have grown up on the UNIX system. Many of these tools may be viewed as an encapsulation of a piece of specialized knowledge--anything from driving a piece of hardware to performing a utility task. They may be as simple as a sin subroutine, or as complex as a disk driver. By using tools, programmers are able to gain access to facilities and algorithms that they do not have to understand in detail. Not surprisingly, constructing programs by using available tools and pieces has proved to be very successful:
>
> - The resulting products are produced quickly.
> - They are likely to work correctly.
> - They are often quite flexible and adaptable as applications change.
> - Inter-machine portability is often enhanced by using tools.
> - Tool usage encourages a natural modularity in the resulting program.
> - Using tools constructed by experts, programmers get the use of expert algorithms.
>
> (Johnson (1980) p. 16)

Because Telecenter consists largely of existing utilities, a quick, low-cost implementation of the system was possible. As new utilities come into existence--for example, a new relational database program or more powerful mail program--it is trival to incorporate them into new Telecenter functions. Utilities are the essence of the system's modular design.

The "programming language" in which Telecenter is written is that of our UNIX system's standard command interpreter program. As a result, its functions consist of easy-to-understand command-interpreter scripts that can be implemented by anyone familiar with the operating system's standard interactive commands. This fact, combined with the interactive manner in which Telecenter scripts can be tested, is what makes the system highly adaptive to user needs. It means individuals with very little programming experience can custom-tailor functions to user needs.

Using high-level, interpretive command-interpreter scripts in Telecenter's early stages does not doom the system to inefficiency. The system's modules all interconnect, regardless of whether they are command-interpreter scripts or compiled programs. The code used to produce a given function is "transparent" and can be modified at a later date if necessary in the name of efficiency. In the meantime, in the first stages of the system's development, user-oriented people--people more interested in what a function does rather than how it does it--are in a position to adapt functions in response to user needs.

Connected with the ease-of-use of the command-interpreter "language" of Telecenter is the file/directory system that permits easy structuring of information on the system. Again, thanks to overall operating-system transparency, it will be possible to replace file/directory organizing mechanisms with more efficient ones when greater efficiency is required.

For a more detailed discussion of these four fundamental aspects of the UNIX operating system that made Telecenter possible, see Appendix II.

CONCLUSION

The phrase "think globally, act locally" was the theme of a recent world futures conference in Canada. It seems to us that these words summarize well what we feel to be of value in the design approach to the UNIX-based computer conferencing system described here. By responding to the needs of actual users, we think that we have created a useful collection of communications functions. By creating them on the technical basis of a standard, widely available and modern operating system such as UNIX, we hope that our approach will be of use to a wider community as a small example of adaptive design in an experimental but useful situation.

APPENDIX I:

As a concise means of reviewing the main features of Telecenter, a sample interaction is presented below in which a user, Lathrop, begins a session. (Words typed by the user Lathrop are underlined, words printed by the computer are not. Miscellaneous comments are placed in square brackets.)

```
login: telectr
Welcome
Name?lathrop
Password?  somepassword
lathrop on at: Thu Jan 29 10:55:30 CET 1981
Last logged in: Thu Jan 29 08:30:08 CET 1981
Conf: eies 3 comments. lathrop has 3 outstanding
Conf. nroff: 17 comments. lathrop has 2 outstanding

To see new items type "seecomments <conference-name>"
To enter new item type "newcomment <conference-name>"
To modify existing item type "modify <conference-name> <comment-number>"
To have an overview of all your conferences type "overview"
To see status of conf. participants type "status <conference-name>"
To see this list type "help"
To reread a comment use "listcomment <conference-name> <comment-number>"
To quit type "--"

##
```

```
    [Lathrop is a member of the conference "nroff" (NROFF is the
name of a text-processing program available under UNIX.) After
checking the status of this conference...]

## status nroff
CONFERENCE: nroff
UP TO 15 (2 unread) lathrop
UP TO 16 (1 unread) mpearson
UP TO DATE          smith

17 ITEMS
##

[Lathrop uses a "seecomments nroff" to see its latest comments;
each new comment is displayed after Lathrop presses a carriage
return.]

## seecomments nroff

Seecomment: Depress carriage return to see next item. Terminate with "--".
Next is comment 16 of 17 in conference "nroff"

[lathrop does a carriage return here]

Conf: nroff          Comment: 16          Lines: 4
Entered by: mpearson   on: Thu Jan 29 09:03:20 CET 1981
Keys: program available for printing comments on the microdot printer

How are things coming on your text editing tutorial?
I have written a short program to simplify sending text
over to the microdot printer.  You can try it the next time you
want to print out your tutorial.

Seecomment: Depress carriage return to see next item. Terminate with "--".
Next is comment 17 of 17 in conference "nroff"

[lathrop does a carriage return here]

Conf: nroff          Comment: 17          Lines: 3
Entered by: smith on: Thu Jan 29 09:55:30 CET 1981
Associated Comment: 12
Keys: standard macro package for IIASA/text-processing

Let's think about making a macro package that anyone at
the institute can use.  There should be one set of input
conventions for all word-processing here.
##

[If Lathrop wants a hard copy of any of the comments in the
"nroff" conference, he or she can use the "lprintcomment" or
"variancomment" commands.  Note that any unambiguous portion of a
command name may be typed.]

## lprint nroff 16
## variancomm nroff 17
##
[Now checking the status of "nroff" shows that Lathrop has seen
everything.]

## statu nroff
```

```
CONFERENCE: nroff
UP TO DATE            lathrop
UP TO 16 (1 unread) mpearson
UP TO DATE            smith

17 ITEMS
```

[Next Lathrop decides to respond to one of the latest "nroff" comments by entering a new comment in that conference.]

```
## new nroff
(You are in the IIASA editor.  Type "x" to use edx.
Type "w" then "q" to enter comment.)
ENTERING SCRATCHPAD:
0
*a
```

[Since Lathrop is using a hardcopy terminal it is necesary to compose and modify text using the append mode of the standard IIASA editor (NOT using the screen editor "edx") and he or she types an "a" in response to the editor prompt, *. This starts the append mode for adding new material to the SCRATCHPAD...]

```
*a
I agree that the institute needs a standard macro package that
anyone here can understand and use.  As far as I know, no such
package has been agreed upon yet.  I suggest we modify Smith's
package --it's relatively simple and straightforward.
.

*w

245

*q
```

[After finishing the comment, by typing a carriage return, a dot and another carriage return, Lathrop now types a "w" (carriage return) and next a "q" (carriage return) to begin the entering process. Telecenter responds with a series of questions that will supply information for the header.]

```
Associated Comment (#)? 17
Keys (Word/Phrase/)?Standard Macro Package
Associated Comment:  17
Keys:  Standard Macro Package
OK to enter (y/n)?
```

[Lathrop is satisfied with the comment and decides to enter it in the conference by typing a "y" in response to Telecenter's latest question.]

```
OK to enter?y
Comment being entered.
Entered as:
Conf: nroff Comment: 18 Lines: 4
Entered by: lathrop on: Thu Jan 29 10:58:10 CET 1981
OK to delete SCRATCHPAD? (y/n)?y
```

[As a final step in entering the comment, Lathrop agrees to delete SCRATCHPAD and types a "y" to do so.]

[Before logging out, Lathrop uses the "terse" command to stop Telecenter from printing out a conference overview and a command list at the beginning of future sessions.]

```
## terse
##
```

Lathrop knows that the omitted information can be obtained with the Telecenter commands:

```
overview
help
```

After this, Lathrop is ready to end the session:

```
## --
login:
```

APPENDIX II:

As mentioned above, four key features of our UNIX system contributed directly to quick implementation of Telecenter. These features are intrinsic to the adaptive design philosophy of the conferencing system.

UNIX utilities

UNIX has many utility programs written to do simple things to text files, like printing lines that contain a specified pattern, eliminating duplicate lines in a sorted file, or printing the first or last few lines of a file. All these simple "tools" are designed to be flexibly combined at the command level to perform more complex tasks, and to use them in this way requires no programming and little effort.

Most of the facilities for doing Telecenter's basic tasks already exist under our UNIX in the form of utilities--for example, a simple screen-oriented editor program edx for creating and modifying text (for users with hardcopy terminals a line-oriented editor is optional,) a utility called grep for searching files for strings of interest, the utility date for inserting the date and time into conference comment headers, notifications of login and logout times, and so forth. They also structure information (see the paragraph below on file system).

Command interpreter

The command interpreter--called the "shell" in UNIX--is the primary mechanism through which time-sharing users interact with UNIX. A shell is a normal, un-privileged user program that typically reads commands typed by the user at his or her terminal and executes functions either through code compiled into the shell itself, by running other programs, or by interpreting text files containing shell commands. The shell used to implement Telecenter is called the C shell and is the shell used by all

UNIX users at our institute (see Joy (1980)). For implementing Telecenter, the C shell was used as a very high level, block structured, interpretive programming language, which provided convenient access to built-in functions, utilities and the various programming features of the C shell itself, including variables, simple numeric expressions, pattern matching, macros and file system manipulation.

Except for one or two trivial compiled programs, all aspects of Telecenter are small (ideally thirty-five to seventy lines of code each) C-shell "programs" (sometimes called scripts) that are easy to write and modify. Since the basic elements of this "programming language" are normal interactive commands, sometimes parametarized and/or conditionalized, powerful functions using many utility programs combined in various ways can be implemented by users with little or no programming experience. Furthermore, since this language is interpretive, it is easy to debug and experiment with programs, or parts of them, merely by typing them interactively at the terminal.

Inter-function transparency

Since there is great consistency between built-in functions, compiled programs and C-shell scripts, all can be combined within C-shell programs or made directly available to the user without knowing or caring how a given function is implemented. This allows modules to be easily replaced by more efficient implementations for efficiency reasons once the required functionality is well-known, presumably determined through a flexible, very high level, interpretive "first attempt".

In a first stage of development, where we are now, appropriate functionality can be explored by user-oriented people able to modify shell-scripts. In a second stage, efficiency can be achieved for those functions deemed desirable by having shell-scripts partially or wholly converted to either compiled programs or functions built into the shell. In the meantime all components of the system are compatible with one another--all modules interconnect. We are confident about the second stage of Telecenter's evolution because UNIX supports a modern, structured programming language called "C" (see Kernighan and Ritchie (1978)) that is superior to FORTRAN and other earlier programming languages and offers the kind of access to the operating system that has been shown to be essential for developing these kinds of systems.[4]

File system

The structuring of information in Telecenter was facilitated by the fact that UNIX has a simple yet powerful file/directory system that supports the organization of files into an arbitrarily deep hierarchy. Every directory can contain files and/or more directories. A directory entry for a file (called a link) can be placed in more than one directory, thus allowing a file to appear in many directories at once.

Organizing Telecenter users, conferences and conference participants was done simply by writing C-shell scripts that manipulate the appropriate directories--a directory hierarchy for users and a directory hierarchy for conferences. Thus when a participant submits a conference comment he or she does so by calling a function that creates and links files in appropriate directories. The Telecenter functions that do these things incorporate standard UNIX utilities for making, moving among, and removing directories as well as those for making, linking, editing and removing files. For example, keeping track of who has seen what is simply a matter of establishing links for each participant to a single comment file, removing a link when someone has viewed a conference comment and then, for any given conference, counting who still has links and how many.

In effect, UNIX's directory/file structure provides a simple database management system for implementors and users of Telecenter.

REFERENCES:

Crowley, T.H., Unix Time-Sharing System: Preface, The Bell System Technical Jrnl. Vol. 57. Number 6, Part 2. July-August 1978, 1897-98.

Gambino, T.J., Implementation Research and DSS. Paper delivered by Gambino on work done together with Peter Keen. Delivered at the 5th Annual Information Systems Summer Workshop held in Williamstown, Mass., U.S.A., whose theme was Strategies for Supporting Management of the Future. (Quoted material on the basis of notes prepared by Alain Barbarie, Management and Technology Area, IIASA.)

Hiltz, S. R., and M. Turoff, The Network Nation. (Addison-Wesley, Reading, Mass. 1978).

Johnson, Stephen C., Language Development Tools on the UNIX System, Computer. Vol. 13, No. 8, 1980, 16-21.

Joy, William, An introduction to the C shell (Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, Berkeley, California 94720, 1980).

Kernighan, B. W. and Ritchie, D.M., The C Programming Language (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978).

McIlroy, M.D., Pinson, E.N, and Tague, B.A., Unix Time-Sharing System: Forward, The Bell System Technical Jrnl. Vol. 57. Number 6, Part 2. July-August 1978, 1899-1904.

Miller, Richard, Representing the Message System User Environment: A Report to the User Environment Sub-Group, IFIP Working Group 6.5. Contribution to International Federation for Information Processing workshop on computer message systems held May, 1980 at the Gesellschaft fuer Mathematik und Datenverarbeitung in Bonn, FRG, and organized by IFIP Working Group 6.5 on international computer message systems. Will be part of a forthcoming workshop proceedings.

Pearson, M.M.L., Using the Computer to Communicate: An Introduction to Computerized Conferencing. WP-80-72. Laxenburg, Austria: International Institute for Applied Systems Analysis. (May 1980).

Pearson, M.M.L. and Lathrop, C.L., Using the Computer to Communicate: A User's Guide to Telecenter. WP-80-109. Laxenburg, Austria: International Institute for Applied Systems Analysis. (June, 1980).

Ritchie, D.M., and Thompson, K., The Unix Time-Sharing System, The Bell System Technical Jrnl. Vol. 57. Number 6, Part 2. July-August 1978, 1905-1928.

Sprague, Ralph H.,Jr., An Overview of Decision Support Systems. Contribution to Task Force meeting on decision support systems held at the International Institute for Applied Systems Analysis (IIASA), Schloss Laxenburg, Laxenburg, Austria on June 23-25, 1980. Proceedings of the Task Force are to appear in Fick, G.P. and Sprague, R.H.,Jr. (eds.), Decision Support Systems: Issues and Challenges (Pergamon Press, Oxford, 1981).

Uhlig, R.P., Farber, D.J., and Bair, J.H., The Office of the Future (North-Holland Publishing Company, Amsterdam, 1979).

FOOTNOTES:

1. UNIX is a trademark of Bell Laboratories. For an excellent introduction to UNIX see The Bell System Technical Journal, Volume 57, No. 6, in particular (Crowly (1978)), (McIlroy, Pinson and Tague (1978)), and (Ritchie and Thompson (1978)).

2. The closest thing in spirit to it with which we are familiar is a proposal by Richard Miller to use Backus Normal Form (BNF) representations (the formalism originally created to describe ALGOL) for functional descriptions of an electronic message system (EMS). Miller stresses the need for a formal description, or at least some consistent set of definitions, of the functions and interactions of electronic message systems, and sees the BNF-type formalism as a starting point for a "common language in which various (electronic message) systems can be conceived, characterized, compared and evaluated." He goes on to say, "A formal representation of the user environment could benefit the EMS field to the same extent that compiler theory 20 years ago made an impact on the rapid development of high-level computer languages."

3. Turoff's EIES system, in particular, has demonstrated how with forgiving software and conscientious on-line consultancy services a system of this kind can acquire a large constituency, including a large number of people who have never used a computer before (see Hiltz and Turoff (1978)). Certain aspects of Telecenter's user interface (the notion of a SCRATCHPAD, for example) are modeled on EIES. The commercial computerized conferencing service, Infomedia, has also achieved success in this area. Infomedia's approach has been to offer very easy-to-use software to its customers (see Planet News, its monthly newsletter).

4. In a review of both hardware and software requirements for implementing computerized conferencing systems, Hiltz and Turoff have pointed out, "The current trend toward insulating the higher-level languages from the operating system needs to be reversed, so that the conference program in that language can control the operating systems. This means that the status and control of interrupts and I/O have to be incorporated at the higher level..." and that "Most conference packages today have to make frequent transitions into the domain of the operating system command language. All user I/O (break key, etc.) has to be removed from the operating system to the conference system so there is no way for the user to get at the operating system." (Hiltz and Turoff (1978) pp. 390-1)

COMPUTER MESSAGE SYSTEM DESIGN

Chairman - D. Twyver

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

THE MESSAGE PROCESSING SYSTEM IN THE DS-1 INTERNATIONAL PACKET SWITCHING SYSTEM

T. NAMIKI, K. SODEYAMA and T. MARUYAMA
Kokusai Denshin Denwa Co., Ltd. (KDD)
3-2,Nishishinjuku 2-Chome
Shinjuku-Ku, Tokyo 160, Japan

ABSTRACT: This paper presents general overview of the message processing subsystem (MPS) which has been recently developed as a part of KDD's public packet switching system. As the most outstanding feature of the MPS, message level protocol (MCP) is developed. The MCP which is constructed on X.25 protocol is designed to interconnect with other message switching centers through packet networks. Access protocols to support various types of terminals are implemented as well. The MPS provides users with variety of message handling facilities including mailbox type function, delayed delivery and delivery certificate.

1. INTRODUCTION

To meet increasing demands for message communications, store and forward type message services are provided by a number of common carriers and remarkable progress has been made during last decade by enhancing services and facilities to be provided. Networks for the message services are also expanding internationally by interconnecting message switching centers which are provided by individual international carriers.
KDD, the international telecommunication carrier of Japan, has inaugurated this type of service called AUTOMEX in 1973. Considering increasing requirements for these message services and progress in packet switched data networks, KDD has developed a message switching system called MPS (Message Processing Subsystem) which constitutes a part of an international packet switching system DS-1. (1) (2) The MPS is designed to operate in the packet switched network and to build an international message switched network in conjunction with overseas message switching centers; for this purpose a message level protocol, MCP (Message Communication Procedure) is developed. The MCP is constructed upon the CCITT X.25 protocol. Capabilities such as mailbox type function, delayed delivery, delivery certificate etc. are also available by the MPS.

2. GENERAL DESCRIPTIONS

2.1 NETWORK CONFIGURATION

In the early stage of AUTOMEX, message services were provided for closed user groups by interconnecting international and domestic leased circuits via store and forward switching system. The system was also connected to overseas message switching centers by leased circuits. Network expansion has been made by connecting domestic telex circuits for message delivery. On the other hand a number of public packet switched networks are being built worldwide. In addition to these conventional connection to the message switching system, application of packet network is aimed in the MPS to interconnect with overseas message switching systems. In consequence, different types of circuits are interconnected by the MPS as shown in Figure 1.

2.2 ADDRESSING

From the network viewpoint, the telex and the packet networks are regarded as subnetworks constituting the message switched network. Corresponding to this

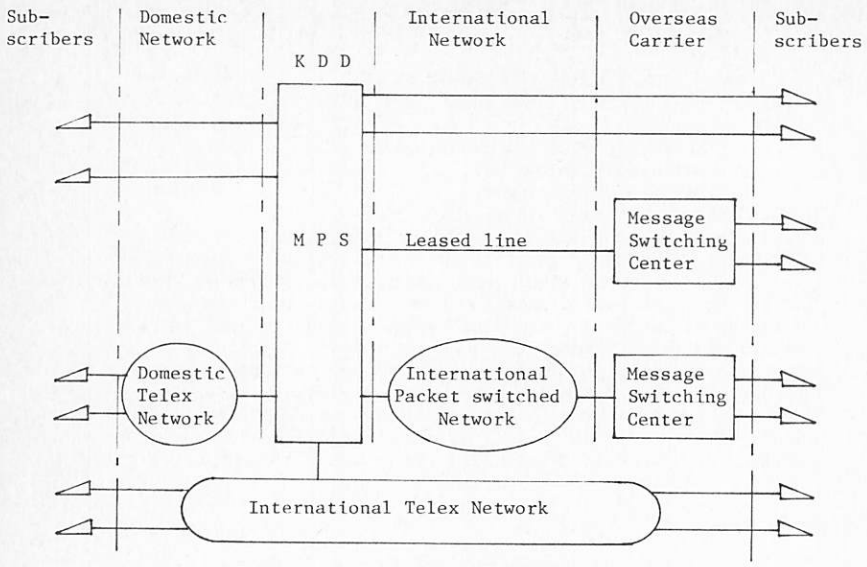
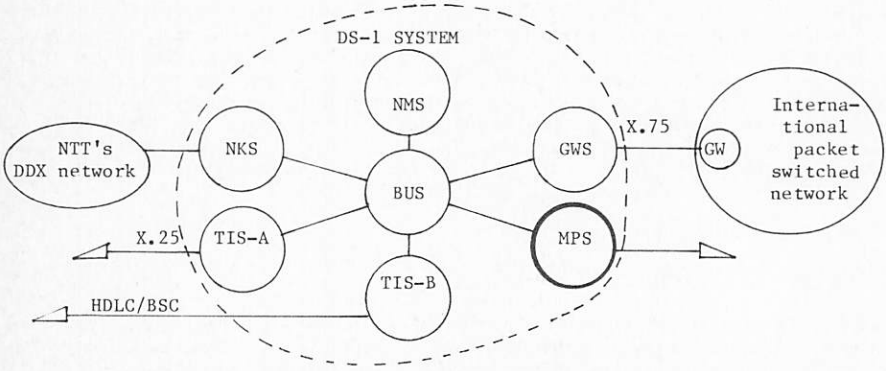


Figure 1 Network Configuration



TIS: Terminal interface subsystem
GWS: Gateway subsystem
NKS: NTT-KDD switching subsystem
MPS: Message processing subsystem
NMS: Network management subsystem
BUS: Bus susbsystem

Figure 2 The DS-1 international packet switching system

structure of the network, two level addressing scheme is applied in the MPS, the message level address and subnetwork address. The message level address is independent of that of the subnetwork and uniquely assigned in the MPS. The address code consists of five or seven characters as described in section 5.1. The subnetwork address is in accordance with its original network. In case of packet network, the address is based on CCITT numbering plan X.121. A subnetwork address (a packet network DTE number) is assigned to each CUG at a message switching center and a message level address is uniquely converted to the subnetwork address.

2.3 SYSTEM DESCRIPTION

The MPS is constructed as a subsystem of the international packet switching system DS-1. The DS-1 system has a building block structure in order to give better expandability and flexibility for the future. Subsystems are equipped corresponding to different types of terminal interfaces or to different services as shown in Figure 2. Each subsystem has a duplex structure for the high reliability.
Hardware configuration of the MPS subsystem is shown in Figure 3. The MPS consists of duplex system of general purpose mini-computer, Panafacom U-400. As for the file equipment, a fixed head disk unit (DKU) and moving head disk units (DCU) are used properly; the DKU is used for storage of system data and non-resident programs, the DCU for retrieval file, mailbox file and delayed delivery file. These units are duplicated and data or messages are always stored in both units for security purpose against losses of them. Capacity of files is shown in Table 2. To realize higher reliability and to minimize the time of single file operation, spare storage units are reserved for shared use in addition to the above mentioned duplicated arrangements.
Diagnostic programs run periodically between on-line system and stand-by system for the detection of abnormal conditions of the system.
If any malfunction is detected, the reconfiguration program checks available possibilities and takes appropriate procedure. This results in shortening repair time which plays an essential part in achieving necessary high system availability.

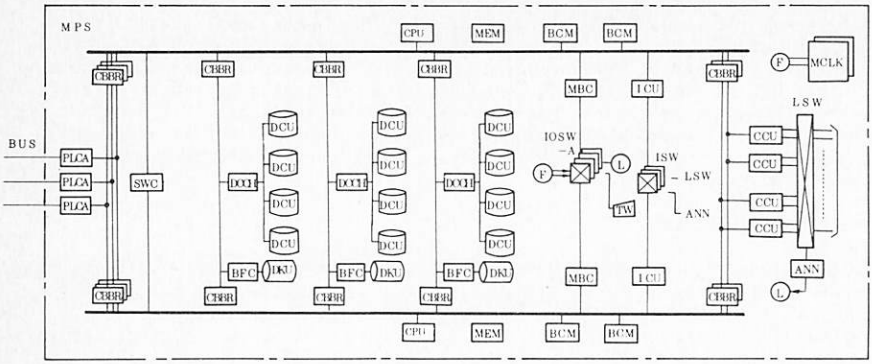
Table 2 File capacity

Name of file	Capacity
Message exchange file	4,200 messages
Message retrieval file	154,800 messages
Mailbox file	10,000 messages
Delayed delivery file	8,000 messages

2.4 SYSTEM OPERATION

As for system operation and maintenance, all subsystems of DS-1 including the MPS are designed so as to be controled and supervised centrally from the system supervisory position of the network management subsystem (NMS). System operation for the addition or alteration of hardware, network parameters etc. is carried out from the supervisory position without causing any interruption of service.
The operation commands for these controls include:

- Control command for the protection of the system data against operation misses.
- System command for the system operations e.g.:
  - initial program loading
  - initiate recovery procedures
  - change hardware configurations
  - inspect and/or change contents of main memory and auxiliary memory.
- Network command for network operations e.g.:
  - introduce or remove CUG



MPS	Message processing subsystem
CPU	Central processing unit
MEM	Main memory
BCM	Bulk core memory
SWC	Switching controller
DCCH	Disk cartridge channel
BFC	Basic file channel
MBC	Multiple basic channel
CCU	Communication control unit
ISW	Interface switching unit
PLCA	Parallel line control adaptor
CBBR	Common bus branch adaptor
ICU	Interface control unit
ANN	Announce machine
DCU	Disk cartridge unit
DKU	Disk unit
IOSW	Input/output switching unit
LSW	Line switching unit
TW	System typewriter

Figure 3. Hardware configuration of MPS

- register and update network parameters
- service request for message retrieval and status report

Some functions of these network operations are left for customers operation as described in the section 5.7.

Table 1 Summary of MPS functions

Item	Function
General Characteristics	Store and Forward Message Switching
Line Speed	50 75 200 1,200 (ASYN) 2,400 4,800 9,600 (SYN)
Protocol	Free Running, F1/H1 polling, H6 polling BSC Polling, Telex, MCP
Network Connection	Leased line, Telex(Dommestic and International) Packet network
Message Exchange	Inter-CUG and Intra-CUG Communication
Message Format	based on ATA/IATA format
Routing	By means of Message Level Address
Code	ITA No.2, IA No.5, EBCDIC
Facilities	Mailbox type function, Delayed Delivery, Delivery Certificate

3. MESSAGE LEVEL PROTOCOL FOR PACKET NETWORK APPLICATION

Packet switched networks are applied in the MPS to interconnect with overseas message switching systems. This enables us to design a reliable and efficient system and to enhance networks effectively. A message level protocol called Message Communication Procedure (MCP) is developed for this purpose. Logically this procedure is constructed upon X.25 protocol. Each message is transfered to the other message switching center by a virtual call which is established for every message. Physically the MPS is connected with the Gateway subsystem (GWS) of DS-1 system through an intra-system protocol which covers all functions of X.25 protocol. Figure 4 shows the logical connection between MPS, GWS and international packet network. Since constructed upon X.25 protocol, the MCP can rely on error-free data transmission of the packet network and rather simple functions are required for the MCP, e.g. virtual call control, message level acknowledgement etc.
These functions and procedures are presented in detail below.

3.1 FUNCTIONS

(1) Virtual call control for message transfer

A message is transfered to the other message switching center by a virtual call which is set up on a message by message basis. Call set-up, data transfer and clearing procedures are performed in accordance with CCITT X.25. The virtual call is cleared after a message level acknowledgement is received. This process is shown in Figure 6.

(2) Message Level Acknowledgement (MLA)

Upon receipt of a message, the MCP verifies the message and send back one of the following three message level aknowledgement (MLA).

- ACK : The message is received in correct format and address codes.
- NAK-0 : The message includes format error and/or invalid address codes.
- NAK-1 : The message is received with wrong message length as compared with the length indicated in the control field of first data packet. This prevents data losses in the message by packet losses.

For establishing this function, a user data control (UDC) field of four octets is provided in the first data packet of a message as shown in Figure 5. The MLA is indicated in the first octet of the UDC field. Message data length is indicated in the last two octets in the UDC field.

(3) Special communication facility

The MCP provides a function to enable special communication through the pcket network. This is indicated by the special number indicator in the first octet of the UDC field (see Figure 5). This function is used in the MPS for communications for network managements.
Message switching center usually has at least one center supervisory position which is generally called XQ position. Center supervisors have to be able to communicate using XQ position with ordinary subscribers of any CUG or supervisors of other message switching centers one another to coordinate various problems associated with their message switching. These communications are treated free of charge.

3.2 PROCEDURE

Procedure for the call set-up/clearing and message transfer is presented in this section. The MPS communicates with overseas message switching systems by the MCP. To simplify the description, the sending node of message switching system is designated as MPS(S) and the receiving node as MPS(R).

a. MPS(S) sends a CR (call request) packet when there is a message to be sent.
b. MPS(R) responds to the CR packet with a CC (call connected) packet when MPS(R) can accept the call.
c. CQ (clear request) packet is sent back when MPS(R) rejects or clears the call because of some reasons, or when one of Timers T1 through T4 expires.
d. MPS(S) starts to send DT (data) packets after having received a CC packet.
e. When MPS(R) has received a message completely, MPS(R) compares the value represented by the message data length indicator in the UDC field with the actual message length.

If the two numbers are equal to each other, the following procedures are taken.

f. MPS(R) sends ACK back to MPS(S) when the validity of the format and address codes of the message has been verified.
g. MPS(R) sends NAK-0 back to MPS(S) when a message is not accesptable because of format error or invalid address codes. A NAK-0 includes a notice indicating a reason why the message is not acceptable. The message having resulted in NAK-0 is intercepted by MPS(S) to the intercept position of the sending CUG with the notice sent back from MPS(R).
h. MPS(S) sends a CQ (clear request) packet after having received an ACK or a NAK-0 in the same virtual call.

If the actual message length differs from that indicated in the UDC field, the following procedures are taken.

i. MPS(R) sends a NAK-1 back to MPS(S). Then MPS(S) tries to retransmit the same message with the same message sequence number as the previous one with PDM (Possible Duplicated Message) indicator. A new virtual call is set up for this message retransmission. The PDM is indicated in the third bit of the first octet of the UDC field.
j. When MPS(S) receives the NAK-1 in response to the retransmitted message three times successively from MPS(R), MPS(S) stops retransmitting the same message and intercepts it to the MPS(S)'s supervisory position.

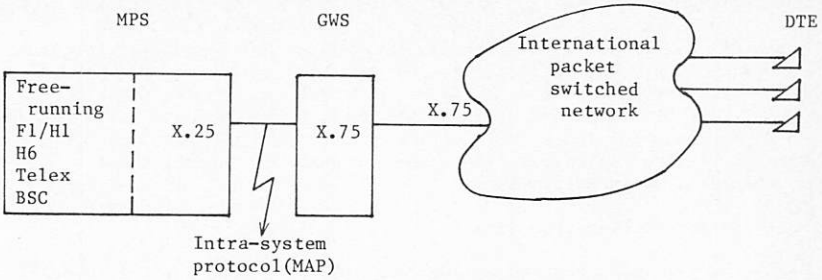


Figure 4 The link between MPS and international packet switched network

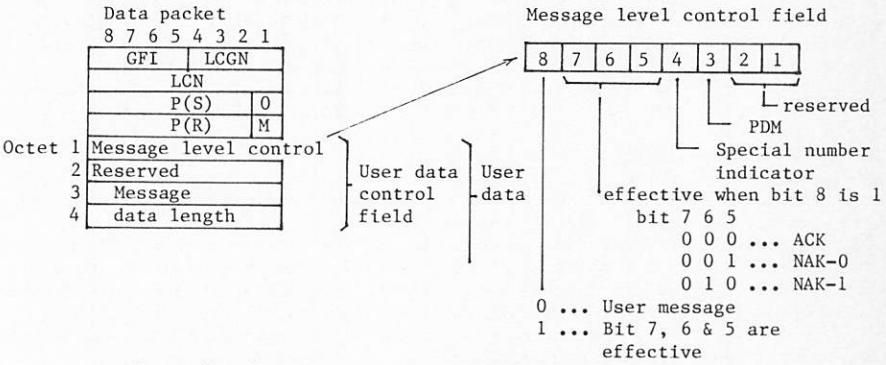


Figure 5 User data control field in the first data packet

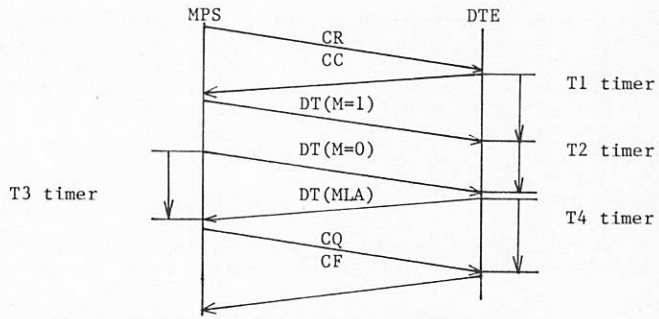


Figure 6 Successful packet transfer procedure

4. TERMINAL INTERFACES

Various types of terminals are supported by the MPS to meet customers' requirements for wide applications. They include low speed teleprinter of ITA No.2 code, asynchronous character terminals with speeds up to 200 bps, medium speed asynchronous CRT or OCR terminals of IA No.5 code with speeds up to 1,200 bps and high speed synchronous terminals (BSC) with speeds up to 9,600 bps either free-running type or polling controlled type. The procedures to support these terminals are briefly presented below.

(1) Full duplex free running procedure

This procedure is for low speed and simple terminals up to 200 bps and no special control functions are required for the terminals. When open-line condition or modem error is detected in the midst of sending message, the transmission of a message is aborted automatically and the message is retransmitted with PDM indicator after the line restoration. In some cases this procedure is applied to intercenter link; in this case a simple control function is added to ensure the message transfer against malfunctions of line or center systems. This procedure called VA/IA CONTROL introduces control-character sequences, VA, IA, and SA to inform of valid state, line failure state and system congestion state respectively. The MPS provides some queues for outstanding messages and manages the queues in accordance with a received sequence, VA, IA, or SA.

(2) Fl/Hl Polling procedure

This procedure is applied to the control of low speed terminals connected to the MPS by a multi-drop circuit of full duplex (Fl) or half duplex (Hl).
Message transmission or reception is controled by the MPS by means of polling or calling sequence. When a message is received from a terminal, the MPS sends a response called IMA(Input Message Acknowledgement) or IMNA(Input Message Negative acknowledgement) as a result of message verification. As no error controls are applied in this procedure, customers have to retransmit a message after correction if necessary in case of IMNA by manual.

(3) H6 Polling procedure

This procedure is applied to asynchronous terminals with transmission speeds up to 1,200 bps and code of IA No.5. To control data links STX, EOT, ACK, NAK and WACK are used.The virtical redundancy check (VRC) is carried out to detect transmission error for both receiving and sending. Automatic retransmission is performed between the MPS and a terminal controller in case of transmission error.

(4) BSC Polling procedure

This procedure is applied to synchronous terminals with transmission speeds of up to 9,600 bps and code of IA No.5 or EBCDIC.
MPS works as a control terminal and user's terminals as tributary. A cyclic redundancy check code, CRC-16A or CRC-16B is selected by user option.

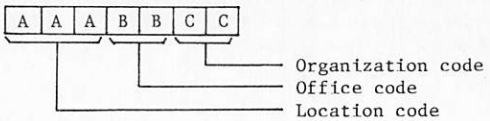
5. MESSAGE HANDLING CAPABILITIES

5.1 MESSAGE DELIVERY CONTROL

(1) Message address

Messages are routed and delivered according to addresses in messages. An address code consists of seven characters by which a location, an office and an organization are represented as shown in Figure 7.

(i) Seven character address



(ii) Five character address

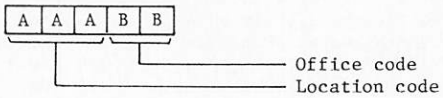


Figure 7 Format of message address code

Two characters of an organization code is uniquely assigned in the MPS to each CUG. The other codes in the address code are freely assigned by each CUG.
In the case of inter-CUG communication a seven-character address code must be used. In the case of intra-CUG communication, however, a five-character address code without an organization code may be used as shown Figure 7.
Abbreviated address is provided to permit shorter encoded address of two to four characters.

(2) Multi-address

Group address code is used to deliver messages to at most 32 destinations by one group address code.
Broadcast address code is also used to deliver messages to at most 100 destinations. The use of this broadcast address code is allowed only to KDD XQ position and customer's TCC position.

(3) Priority

Four priorities are provided for message delivery. Customers are allowed to select one of the lower three priorities for each message. The highest priority is provided for the messages generated by the MPS.

5.2 SECURITY ARRANGEMENTS

The loss of messages is protected by checking sequence numbers of input and output messages. Input sequence numbers of messages are checked by the MPS and output sequence numbers are put to outgoing messages from MPS to every terminal.
If an irregularity in the input sequence number is detected by the MPS, a notice indicating sequence number error is generated and forwarded to a predetermined terminal. By this error notice, user can take an appropriate action to avoid message losses. These sequence numbers are also used as reference numbers for message retrieval as described in section 5.6. In case of packet network connection with foreign message switching centers, messages are further secured by the message level acknowledgement (MLA) of the MCP as described in section 3.1.

5.3 DELIVERY CERTIFICATE

The MPS generates a delivery certificate notice to the originator of a message after the message has been delivered to its recipient. The delivery certificate notice is generated only when this facility is requested by a special priority

indicator in a message. The destination of the notice can be specified by an originator indicator in the original message. The MPS does not issue a delivery certificate notice if the message is sent to another message switching center, because it should be genetated by the center nearest to the recipient as an end-to-end acknowledgment.

5.4 DELAYED DELIVERY

The MPS has a function of holding messages in the delayed delivery file for a period of time on recipient's request basis. Messages addressed to a terminal which requests this service by specifying a delivery time using user's command are stored in the delayed delivery file and they are delivered at the time specified in advance. The stored messages are also delivered when the release command is entered or the file becomes full.

5.5 MAILBOX TYPE FUNCTION

The MPS provides mailboxes and enables users to send and receive confidential messages. Each mailbox has a personal password, known only to the person to whom the mailbox is assigned to use. A user may change his password any time by the user's command. When the MPS receives messages having mailbox addresses, the MPS stores and keeps them in the mailbox file in the MPS in stead of sending directly. The recipient can request to pick up the messages in his mailbox at any time from any terminal of his CUG by read command with his personal password. The mailed messages can be transfered to any other terminals which is specified by the read command.

5.6 MESSAGE RETRIEVAL AND TRANSFER

The MPS provides message retrieval and transfer services. All messages having been sent from the MPS are stored in the message retrieval file. These messages in the file can be retrieved or transfered to destinations which are indicated in the message retrieval commands. The messages to be retrieved are designated by their input/output message sequence numbers or the estimated time when the original massages were delivered. DCU is used for the retrieval file so that the retrieval service is automatically carried out by user's commands. The file has a capacity of storing messages for a period of approximately four days for the muximum traffic after transmission from the MPS with full duplicated file. Retrieval file assignment for each CUG is designed to be flexible. If a CUG needs longer period of message storage, it is possible to assign its file single.

5.7 NETWORK MANAGEMENT BY CUSTOMERS

Most facilities are left to users to operate user's network, e.g. supervision of circuit and termianl status, delivery control of messages and error processing of the intercepted message. For this purpose one of some supervisory positions called Traffic Control Center (TCC) are provided. There are two kinds of supervisory positions. One is for alarm and intercept facilities and the other is for command and response. The facilities left to user's operation for network control are as follows.

(1) Message interception

The MPS intercepts invalid messages to a TCC position or an originating terminal of the messages. The destination of intercepted messages is determined by user's terminal option. By this option the user can select to process invalid messages either at TCC position in bulk or at each original terminal.

(2) Network management and service request

Following functions are provided for customers.

- Network management
  - Supervision of alarm information on network and traffic
  - Control of output hold and its clear
  - Control of output alternation

- Service request
  - Control of delayed delivery
  - Message retrieval request
  - Message transfer request
  - Status request on circuits, ternamals, files, message queues and message sequence numbers
  - Registration or change of password of mailbox

6. CONCLUSION

In this paper a message switching system which has a capability to accesss to/from a packet switched network is presented. In the system, the message communication procedure is introduced to operate on the X.25 packet network, and its effectiveness has been described to realize reliable message communication with rather simple implementation.

REFERENCES

1. Ichihara, H., "Planning for the international data network service - VENUS project" Proc. ICCC '78,pp. 57-62, 1978.

2. Ohyama, N., Inoue, S., Nakajo, T., "DS-1 packet switching system for new international public data transmission service VENUS" ISS '79, pp. 447-453, 1979.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

SERVICE DEFINITIONS in a COMPUTER BASED MAIL ENVIRONMENT

Peter Schicker

Zellweger Uster AG
CH-8634 Hombrechtikon
Switzerland

List of services to be provided by computer based mail and message systems are often compiled by editing lists of existing communication services (e.g., post, telephone, telex, etc.). Unfortunately, these lists seldom specify how the service shall be adapted to the new environment and misunderstandings and ambiguities are the natural consequence of the inprecision.

On the example of a very small class of such services it is shown how verbal and (semi-)formal definitions of all the different aspects of a service can lead to a common understanding. This understanding is the basis from which user communities, system designers and standardization bodies can select services for the computer based mail and message systems.

1. INTRODUCTION

Experimental and prototype Computer Based Mail and Messaging Systems have been around for several years. More and more attention and effort is devoted to such systems and today various systems offer their services on a commercial basis. Furthermore, an activity was started within CCITT [1] which will ultimately lead to standardization in this area.

Users of such computer based mail systems are less intrigued by the various internal mechanisms and resource allocation strategies but require exact definitions of the facilities and services that these systems offer. Prospective users, system designers, and service offering companies often compile lists of potential services, e.g., like the list shown in appendix A. Nobody claims that these lists are complete and most often it is admitted freely that these lists represent a first cut synthesis of services offered by other communication facilities (e.g., postal service, telephone, telegraph, telex, etc.).

Unfortunately, these lists mostly convey just a number of buzz-words which everybody interprets in his own fashion. For example, a multitudiness of shades of different services can all be catalogued under 'delivery confirmation' and 'registered mail'. Recognizing this diversity and considering that verbal definitions still can be interpreted in different ways, we need first a tool for formal definition and second employ this tool for the definition of the services.

Because it is not inherently clear which variation of a service is the most useful for the potential users we must produce exact definitions of many possible shades of these services. Only when a common unambiguous understanding of the services is reached can user communities, system designers, and standardization bodies evaluate the different possibilities.

This paper shows several steps on the path to this common comprehension. A model of the computer based mail system [2,3,4] is used as a starting point for the definition of the services (chapter 2 summarizes this model). The group of the monitoring services (c.f. appendix A) has been chosen as an example and in chapter 3 a list of verbal definitions of progressive shades of confirmation services is given. Finally, in chapters 4 and 5 examples of semi-formal definitions of some of the services are presented.

2. THE MODEL OF THE COMPUTER BASED MAIL ENVIRONMENT

In order to describe services within a computer based mail system it is necessary to divide the system into a set of functional entities. Let us start from the mail item, i.e. the unit of information that is exchanged between mail system users. There exist three fundamental operations on the mail item: the creation and preparation of the mail item by the originator, the transfer by the mail system, and the interpretation by the recipient. These three operations serve as a guideline for the partitioning of the mail environment (Fig. 1).

The originator prepares a mail item within a functional entity called user agent. It is anticipated that many different implementations of user agents will exist all of them tailored to the specific needs of their respective users. For example, a user agent specialized to handle facsimile messages certainly will be different from a user agent that manipulates personal messages.

Once a mail item is prepared, the user agent initiates a posting dialogue with the mail system. Upon completion of this dialogue, the responsibility for the mail item lies with the mail system that now transfers the mail item towards the destination. There, the recipient's user agent is engaged in a dialogue for the delivery of the mail item. After delivery, the recipient's user agent assumes responsibility for the mail item and the recipient can interpret the content of the mail item. The transfer points of responsibility for mail items between user agents and the mail system are closely analogous to the mail slots in the postal services; hence, the transfer points have been named 'slots'.

For many purposes, the internal structure of the mail system is of little interest. However, where necessary, the functional entity 'mail system' is sub-divided into smaller functional entities (mail system agents). Relating to their particular function when transferring an individual mail item they are often labelled 'posting mail system agent', 'relaying mail system agent', and 'delivering mail system agent'.

The mail item is also structured into two parts: the envelope and the content. The mail item content represents essentially the information that the mail system users want to transfer through the mail system. For the mail system this information is transparent, i.e., it is never used for any mail system internal function nor is it altered on its path from the posting slot to the delivery slot. On the other hand, the mail item envelope contains all information needed by the mail system to forward and deliver the mail item; information to invoke mail system services is recorded in the envelope as well.

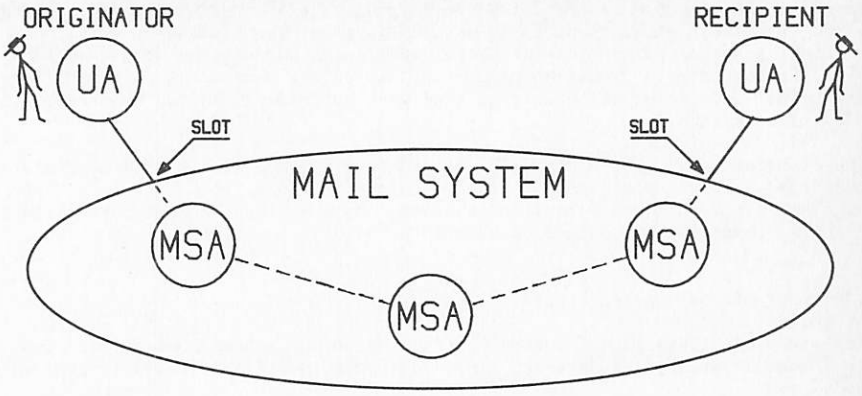


Figure 1

Functional Entities in the Model of the Computer Based Mail Environment

3. VERBAL DESCRIPTION OF THE MAIL SYSTEM MONITORING SERVICES

The mail system monitoring services have been chosen to illustrate the many possible facets of services that all fall under the heading 'delivery confirmation' and 'registered mail'. Several of the services listed below have been suggested to the author during discussions; only a few extra variations have been added by the author in order to complete the list.

The number of items on this list shows clearly that precise service definitons are an absolute requirement; just two unqualified service descriptions ('delivery confirmation' and 'registered mail') give reason for twelve different services. In addition, should these services be discussed within a working group, it is to be expected that a further number of possible variations are unearthed.

The intent of the monitoring services is to provide the originator with information about the outcome of a transfer of a mail item through the mail system. The amount of information that can be conveyed to the originator is bounded by the fact that the delivery slot constitutes an information barrier that is under control of the recipient. In other words, once a mail item is delivered the mail system supported monitoring stops. Further monitoring information (e.g., whether a recipient has read the mail item) is only possible via a concerted action between user agents, but such actions are no mail system services and, therefore, not listed in this chapter.

The monitoring services are not intended to satisfy an originator's curiosity. They are designed to allow the originator to correlate actions outside the mail system with the fact that a particular mail item has been delivered to the recipient and thus anticipating a corresponding correlation by the recipient. Such correlations can lead to disputes, though, and the originator might need to prove to a third party (e.g., a judge) that a particular mail item has indeed been delivered.

Mechanisms for proofs (arbitrated by the mail system [8]) are discussed in the next chapter. The rest of this chapter focuses on a verbal description of a set of monitoring services that intend to give progressive information of delivery confirmations.

As already mentioned in the introduction, the list of services are not intended to be available simultaneously in any mail system. The purpose of this paper is to show by way of an example how verbal and formal definitions of many shades of a particular service must be prepared to allow the assessment of the different variations and the ultimate selection of particular services for standardization and implementation.

Time-stamping and the assignment of globally unique mail item identifiers are considered to be basic and necessary tools to perform the monitoring. They require similar verbal and formal definitions; however, they are not defined in the following chapters but assumed to be available [5].

3.1. Proof of posting (envelope)

This service allows an originator to prove to third parties that a mail item has been posted at a given date and time. The assertion is only valid for the envelope.

3.2. Proof of posting (envelope and content)

This service allows an originator to prove to third parties that a mail item has been posted at a given date and time. The assertion is valid for the complete mail item, i.e., envelope and content.

3.3. Delivery confirmation

This service returns to the originator a confirmation and the date and time of the delivery of the mail item to the recipient's user agent. In case the delivery fails an indication of the reason for the failure and the date and time of it is returned to the originator. Only the mail system's concept of delivery is affirmed, i.e., the originator can not deduce save arrival of the mail item at the recipient's user agent.

3.4. Delivery confirmation with proof (envelope)

This service is an augmentation of 3.3 (delivery confirmation). It allows an originator to prove to third parties the successful or attempted delivery of a mail item. The assertion is valid only for the envelope.

3.5. Delivery confirmation with proof (envelope and content)

This service is an augmentation of 3.3 (delivery confirmation). It allows an originator to prove to third parties the successful or attempted delivery of a mail item. The assertion is valid for the complete mail item, i.e., envelope and content.

3.6. Confirmation of receipt

This service is based on the recipient's user agent acknowledging the receipt of a mail item. The positive or negative acknowledgement is conveyed to the originator. A negative acknowledgement can for example indicate that the recipient's user agent is unable to process the content (e.g., unknown protocol).

3.7. Confirmation of receipt with proof (envelope)

This service is an augmentation of 3.6 (confirmation of receipt). It allows an originator to prove to third parties the acceptance or rejection of a mail item by the recipient. The assertion is valid only for the envelope.

3.8. Confirmation of receipt with proof (envelope and content)

This service is an augmentation of 3.6 (confirmation of receipt). It allows an originator to prove to third parties the acceptance or rejection of a mail item by the recipient. The assertion is valid for the complete mail item, i.e., envelope and content.

3.9. Notification of receipt

The recipient or its representative acknowledge their awareness of the delivery of a mail item. The mail item is delivered only after the mail system receives this acknowledgement. If the notification of the recipient or its representative about the pending delivery is not acknowledged within a certain time limit, the mail item is to be considered undeliverable.

3.10. Notification of receipt with proof (envelope)

This service is an augmentation of 3.9 (notification of receipt). It allows an originator to prove to third parties the recipient's (or its representative's) awareness about the delivery of the mail item. The assertion is only valid for the envelope. Note that this service comes closest to the registered mail service of the postal system.

3.11. Notification of receipt with proof (envelope and content)

This service is an augmentation of 3.9 (notification of receipt). It allows an originator to prove to third parties the recipient's (or its representative's) awareness about the delivery of the mail item. The assertion is valid for the complete mail item, i.e., envelope and content.

3.12. Recipient's acknowledgement

The recipient or its representative acknowledge the content of the mail item. This service is clearly beyond the capabilities of the mail system and falls into the category of concerted user agent actions and not into the mail system services.

4. MAIL SYSTEM PROOFS

Several of the services mentioned in chapter 3 indicate the possibility of a proof. Such proofs are not absolute because they rely on the trustworthyness of the mail system and, therefore, belong to the category of arbitrated proofs. They affirm (or negate) that, according to the mail system's knowledge, a particular mail item was posted, delivered, etc.

It is not envisaged that there will exist malicious mail systems that deliberately alter mail items or falsify proofs to favour a particular user. However, users can protect themselves against unsolicited modifications by employing encryption techniques [6]. On the other hand, if a mail system for example confirms delivery

of a mail item without even attempting a delivery, the mail system does not fulfill its contractual obligation to the originator; such behaviour will most likely lead to a separate dispute.

Within the limits of the mail system's trustworthyness for establishing proofs, the following chapters describe three alternative algorithms: the copy, the hashing, and the signature algorithm (the abbreviations and notation used in the semi-formal definitions correspond to those described in appendix B). Again, the three different algorithms show that an unqualified reference to a service 'arbitrated proof' can be interpreted many ways. Although all three algorithms satisfy the service requirements, the service rendered and the implications of how to use the service is different in all three cases. For example, the copy algorithm does not require that the originator maintains a copy of the mail item; the other two algorithms depend on copies supplied by the originator (or the judge) for executing the proof. On the other hand, the copy and the hashing algorithms require that the mail system is equipped with permanent storage in order to execute a possibly necessary proof.

Note that small alterations to these algorithms would also allow a recipient to prove to third parties that a particular mail item from a specific originator has been delivered to him.

4.1. The copy algorithm

The basis of the proof in this algorithm is a copy of the mail item that the mail system keeps in its permanent storage. When asked for a proof, the mail system extracts the mail item from its store and returns it to the requestor of the proof.

Several detail problems have to be considered before the algorithm becomes useful:

- the originator and the mail system must agree on a reference to specify a particular mail item.
- safeguards must be implemented such that only persons authorized by the originator can extract a mail item.

Because references to a mail item must be unique within a mail system it is proposed that the mail system generates a globally unique mail item identifier that is returned to the originator during the posting dialogue. Subsequently, this identifier must be submitted when requesting a proof.

However, a mail system user that is interested in other users' mail items might generate random identifiers and try to extract mail items; thus he might get insight into as many mail items as he desires. If the invader has some additional knowledge about the algorithm by which the mail system generates mail item identifiers his search for particular information can become very precise. To prevent such invasion it is proposed that the mail system generates with a secret algorithm a random bit string that is returned to the originator. Evidently, precautions must be undertaken to assure that the set of all possible bit strings is very sparsely populated. When requesting a proof, the originator then must supply the bit string to the mail system.

The semi-formal definition of the copy algorithm looks as follows:

Posting:

```
UAO->MSAP: POST ( rec,cont )                                    (4.0)
           returns ( s ) ;

MSAP:      MSAmid := GMID ;                                     (4.1)
           h := HASH ( MSAmid,org,rec,cont ) ;                  (4.2)
           s := DMSAP ( h,MSAmid ) ;                            (4.3)
           put ( MSAmid,org,rec,cont ) into                     (4.4)
               permanent storage ;
           return ( s ) ;                                       (4.5)
```

Proof:

```
UA->MSAP:  PROOF ( s )                                          (4.6)
           returns ( pr ) ;

MSAP:      (h,MSAmid) := EMSAP ( s ) ;                          (4.7)
           use MSAmid to extract (MSAmid,org,rec,               (4.8)
               cont) from store ;
           h1 := HASH ( MSAmid,org,rec,cont ) ;                 (4.9)
           pr := (h=h1) ;                                       (4.10)
           return ( pr ) ;                                      (4.11)

MSAP->UA:  if pr then                                           (4.12)
               DELIVER ( MSAmid,org,rec,cont ) ;
```

Notes:

(4.3) The secret algorithm lies with the secret signing algorithm of the mail system agent.

(4.4/8/12) Any other relevant information can be included (e.g., delivery confirmation, time-stamps, etc.).

4.2. The hashing algorithm

When transferring a mail item the mail system executes a hash function on the mail item. Only the result of the hash function is kept in the mail system's internal permanent storage thus greatly reducing storage requirements in comparison with the copy algorithm.

A user who wishes to exercise a proof must submit the complete mail item to the mail system again. The mail system executes the same hash function again and tries to match the result of the hash function with values kept in its permanent store. The result of this look-up is returned to the requestor of the proof. In other words, the mail system affirms or negates to the requestor of the proof that the mail item submitted has been transferred through the mail system.

The semi-formal definition of the proof then looks as follows:

Posting:

```
UAO->MSAP: POST ( rec,cont )                                    (4.20)
           returns ( MSAmid ) ;

MSAP:      MSAmid := GMID ;                                     (4.21)
           return ( MSAmid ) ;                                  (4.22)
           h := HASH ( MSAmid,org,rec,cont ) ;                  (4.23)
           put ( MSAmid,h ) into permanent store ;              (4.24)
```

Proof:

```
UA->MSAP:  PROOF ( MSAmid,org,rec,cont )                        (4.25)
           returns ( pr ) ;

MSAP:      h := HASH ( MSAmid,org,rec,cont ) ;                  (4.26)
           pr := ( MSAmid,h ) in store ;                        (4.27)
           return ( pr ) ;                                      (4.28)
```

Notes:

(4.20) A globally unique mail item identifier that is generated by the mail system is again proposed as a common reference.

(4.23/25/27) Any other relevant information may be included as well (e.g., delivery confirmation, time-stamps, etc.).

4.3. The signature algorithm

This algorithm relies completely on public encryption techniques. When transferring a mail item, the mail system executes a publicly known hash function on the mail item (reduction of information). The result of the hash function is further subjected to the secret signing algorithm of the mail system and only the signature is returned to the originator. No information is retained in internal storage.

The proof is conducted completely outside the mail system. A user wishing to exercise a proof subjects the mail item, of which he must have kept a copy, to the same publicly known hash function. He also executes the publicly known algorithm to verify the mail system's signatures on the signature that was received from the mail system during the posting dialogue. If the two operations yield the same result the proof affirms that the mail system has transferred the mail item in question.

The semi-formal definition of the proof then looks as follows:

Posting:

```
UAO->MSAP: POST ( rec,cont )                                    (4.30)
           returns ( s ) ;

MSAP:      h := PHASH ( org,rec,cont ) ;                        (4.31)
           s := DMSAP ( h ) ;                                   (4.32)
           return ( s ) ;                                       (4.33)
```

Proof:

```
UA:       h1 := PHASH ( org,rec,cont ) ;                    (4.34)
          h2 := EMSAP ( s ) ;                               (4.35)
          pr := (h1=h2) ;                                   (4.36)
```

Notes:

(4.31/34) Any other relevant information can be included (e.g., delivery confirmation, time-stamps, etc.).

5. SEMI-FORMAL DEFINITIONS

The verbal definitions for the confirmation services given in chapter 3 still lack the precision necessary to assure unambiguity. Unfortunately, the world of protocol and service designers still are waiting for a universally accepted and generally understood formalism for accurate specifications. Therefore, we revert to a semi-formal definition that for the moment renders enough precision for the evaluation of potential future mail system services.

As stated in the introduction, all possible services require an exact definition such that they can be evaluated with respect to user requirements, implementation effort, and resource requirement. However, considering the sometimes minute differences between the various services such a list is very repetitious. As the emphasis of this paper is to show a path towards the selection of services rather than the proposal of detailed services, only two examples will be presented. The abbreviations and notations used in the semi-formal definitions correspond to those described in appendix B.

5.1. Delivery confirmation with proof (envelope and content)

The proof mechanism imbedded in this service definition corresponds to the signature algorithm (c.f. chapter 4.3). Similar definitions, deviating only in a few statements, can be given when other algorithms for proofs are employed.

Posting and Delivery:

```
UAO->MSAP: POST ( UAmid,rec,cont )                          (5.0)
           returns ( MSAmid,stamp1 ) ;

MSAP:      stamp1 := TIME ;                                 (5.1)
           MSAmid := GMID ;                                 (5.2)
           return ( MSAmid,stamp1 ) ;                       (5.3)

MSAP->MSAD:RELAY ( UAmid,MSAmid,org,rec,                    (5.4)
               stamp1,cont ) ;

MSAD:      stamp2 := TIME ;                                 (5.5)

MSAD->UAR: DELIVER ( MSAmid,org,rec,                        (5.6)
               stamp1,stamp2,cont ) ;

MSAD:      h := PHASH ( MSAmid,org,rec,                     (5.7)
               stamp1,stamp2,cont ) ;
```

```
MSAD->MSAP:MSACONF ( UAmid,stamp2,h,org ) ;                    (5.8)

MSAP:      s := DMSAP ( h ) ;                                  (5.9)

MSAP->UAO: CONF ( UAmid,stamp2,s ) ;                           (5.10)
```

Proof:

```
UA:        h1 := PHASH ( MSAmid,org,rec,                       (5.11)
               stamp1,stamp2,cont ) ;
           h2 := EMSAP ( s ) ;                                 (5.12)
           pr := (h1=h2) ;                                     (5.13)
```

Failure Notification:

```
MSAD:      stamp2 := TIME ;                                    (5.14)
           reason := .... ;                                    (5.15)
           h := PHASH ( MSAmid,org,rec,reason,                 (5.16)
               stamp1,stamp2,cont ) ;

MSAD->MSAP:MSAFAIL ( UAmid,stamp2,h,reason,org ) ;             (5.17)

MSAP:      s := DMSAP ( h ) ;                                  (5.18)

MSAP->UAO: FAIL ( UAmid,stamp2,s,reason ) ;                    (5.19)
```

Proof:

```
UA:        h1 := PHASH ( MSAmid,org,rec,reason,                (5.20)
               stamp1,stamp2,cont ) ;
           h2 := EMSAP ( s ) ;                                 (5.21)
           pr := (h1=h2) ;                                     (5.22)
```

Notes:

(5.0) This procedural notation defines the information exchange in the posting dialogue. The parameter 'UAmid' is an originator supplied mail item identifier that later is copied into return information to the originator (5.10 and 5.19) and allows the originator to correlate replies to mail items posted.

(5.4) For the definition in this chapter no intermediate mail system agents are assumed; the relaying is directly performed by the posting and delivering mail system agents. This simplification does not influence the generality, though, as the presence of intermediate maisystem agents would be reflected by iterations of statements (5.4), (5.8), and (5.17).

(5.4) If the delivering mail system agent is identical with the posting mail system agent this statement is omitted; likewise, statements (5.8) and (5.17) are omitted, too.

(5.5) According to the verbal definition, the time of delivery or failure must be returned to the originator.

(5.7/9) The functions PHASH ans DMSAP are not necessarily executed in the same mail system agent. The reason for performing the function DMSAP in the posting mail system agent is to permit different signing algorithms for each mail system agent; yet, the mail system user need know only the verifying algorithm of the mail system agent with which he posts his mail items. As by the definition of the service the mail item

content must be included in the parameters to the signing algorithm, the content must be either remembered by the posting mail system agent, returned to it, or the information reducing function PHASH is executed in the delivering mail system agent and only the reduced bit string is returned to the posting mail system agent. As the last alternative consumes the least resources it has been chosen for this definition.

(5.6) The service confirms only the mail system's conception of delivery; there is no reply from the recipient's user agent involved in the confirmation.

5.2. Notification of receipt

This is a second example of a semi-formal definition of a service and has been chosen to be presented in this paper because it shows some limitations of the computer based mail environment that must be observed carefully when defining services. In this case, the limitation stems from the partitioning of the model into its functional entities and can be characterized by the fact that the mail system can not reach the recipient directly but has to interact through the recipient's user agent. However, the models division into user agents and mail system is primarily to allow a multitude of user agent implementations, private and public, each of them closely tailored to their respective users' requirements. This in turn indicates that only the communication behaviour across the slot can be inforced by the mail system; any user agent internal mechanism is beyond the control of the mail system (see notes below).

It is important to notice that other models of the mail environment experience limitations, too. For example it could be argued that the prohibition of private user agents would remove the limitation mentioned in the paragraph above. However, the mail system still is not sure whether it reaches the intended recipient as the recipient can disclose information to other people such that they can pose as him. The particular problem seems to be solvable only when certified thumb print recognition or similar devices are employed.

Posting and Delivery:

```
UAO->MSAP: POST ( UAmid,rec,cont )                                  (5.30)
           returns ( MSAmid,stamp1 ) ;

MSAP:      stamp1 := TIME ;                                         (5.31)
           MSAmid := GMID ;                                         (5.32)
           return ( MSAmid,stamp1 ) ;                               (5.33)

MSAP->MSAD:RELAY ( UAmid,MSAmid,org,rec,                            (5.34)
               stamp1,cont ) ;

MSAD:      stamp3 := TIME ;                                         (5.35)

MSAD->UAR: NOTIFY ( MSAmid,org,stamp3 ) ;                           (5.36)

UAR->MSAD: GET ( MSAmid ) ;                                         (5.37)

MSAD:      stamp2 := TIME ;                                         (5.38)

MSAD->UAR: DELIVER ( MSAmid,org,stamp1,stamp2,                      (5.39)
               cont ) returns ( status ) ;
```

```
UAR:        status := accepted ;      or                          (5.40)
            status := rejected ;                                  (5.41)
            return ( status ) ;                                   (5.42)

MSAD->MSAP:MSACONF ( UAmid,stamp2,status,org ) ;                  (5.43)

MSAP->UAO: CONF ( UAmid,stamp2,status ) ;                         (5.44)
```

Notes (see also notes in chapter 5.1):

(5.36) It is anticipated that the notification is brought to the recipient's attention who actively triggers the execution of statement (5.37). This behaviour, being in the realm of the user agent environment, can not be inforced and user agent implementations might bypass the involvement of the recipient. However, the verbal definition (c.f. chapter 3.9) is quite explicit in allowing a recipient's representative to act on behalf of the recipient. Thus, a user agent bypassing the recipient acts on the recipient's behalf and is therefore its representative.

(5.36) The only required parameter in this statement is 'MSAmid' which must be copied into statement (5.37). The parameter 'org' conveys information about the origin of the mail item and 'stamp3' allows the recipient or its representative to evaluate the time remaining before the delivering mail system agent times out and lets the mail item fail.

6. SUMMARY

It has been shown that computer based mail system's services that are derived from intuitive adaptations of telephone, telex, postal, or other services lend to a multitude of interpretations of the services' mechanisms. In order that user communities, system designers, and standardization bodies arrive at a common understanding and can determine which services to implement, the various shades of all services must be described verbally and formally.

The algorithms presented in the previous chapters show on the example of 'delivery confirmation' and 'registered mail' services the possible break-up into many different shades of the service. Although only a few of the verbally described services are defined formally, ultimately, each verbal definition of the different shades of the services must be accompanied by the formal definition in order to promote the common unambiguous understanding of the services' mechanisms.

7. ACKNOWLEDGEMENTS

The semi-formal definition technique used in this paper is derived from a method developed at the fourth meeting of the North-American Systems Environment Subgroup of the IFIP working group WG 6.5 [7]. Substantial modifications to an earlier draft version of the paper have been suggested by the participants of the working subgroup mentioned above and especially by Ian Kerr and Ian Cunningham of Bell-Northern Research.

8. BIBLIOGRAPHY

[1] 'Message Handling Facilities (Geneva 1980)', CCITT Study Group VII, AP 7 #12 Question 5/VII

[2] P. Schicker, 'The Computer Based Mail Environment: an Overview', December 1979, submitted for publication

[3] 'Report of the 2-nd Meeting of the System Environment Sub-Group', September 1979, IFIP WG 6.5 N16

[4] 'Report of the 3-rd Meeting of the System Environment Sub-group', December 1979, IFIP WG 6.5 N17

[5] P. Dickson, 'Time-Stamping in Computer Based Message Systems', June 1980, IFIP WG 6.5

[6] P. Schicker, 'Seals and Signatures in a Computer Based Mail Environment', November 1979, submitted for publication

[7] 'Report of the 4-th Meeting of the System Environment Sub-group', April 1980, IFIP WG 6.5 N25

[8] D.W. Davies and W.L. Price, 'The Application of Digital Signatures based on Public Key Cryptosystems', 5th International Conference on Computer Communication, pp 525..530, October 1980, North-Holland, Amsterdam

APPENDIX A: SUMMARY OF MAIL SYSTEM SERVICES

Monitoring
* global mail item identifier
* time stamps
* delivery confirmation
* non-delivery notification
* registered mail

Delivery
* time delivery
* special delivery specifications

Distribution / Routing
* addressing options
* urgency classes
* source / hybrid routing
* broadcast / distribution lists
* bagging

Security
* closed user groups
* mail item cancellation / decay
* authentication

Money
* accounting
* reverse charge
* charge on delivery
* reply paid

APPENDIX B: ABBREVIATIONS USED IN THE DEFINITIONS

For the semi-formal definition, the following abbreviations and designations have been used:

UA	any user agent
UAO	the originator's user agent
UAR	the recipient's user agent
MSAP	the mail system agent that performs the posting dialogue (i.e., the posting mail system agent)
MSAD	the mail system agent that performs the delivery dialogue (i.e., the delivering mail system agent)
org	the mail system's identification of the originator (i.e., slot identifier)
rec	the mail system's identification of the recipient (i.e., slot identifier)
cont	the content of the mail item
UAmid	a mail item identifier that is assigned by the originator or its user agent
MSAmid	a globally unique mail item identifier assigned by the mail system agent that performs the posting
stamp1	a time stamp assigned by a mail system agent
stamp2	a time stamp assigned by a mail system agent
stamp3	a time stamp assigned by a mail system agent
pr	the outcome of a proposition in a proof, i.e., true or false
s,h,h1,h2	bit sequences
reason	a reason for a failure to deliver
status	detailed description of the result of the delivery

The following functions are used for the definitions without further specifications:

TIME	generates a time-stamp
GMID	generates a globally unique mail item identifier
HASH	a not necessarily publicly known hash function; if several arguments are supplied they are concatenated before the hash function is applied
PHASH	a publicly known hash function; if several arguments are supplied they are concatenated before the hash function is applied
DMSAP	the secret signing algorithm of the mail system agent
EMSAP	the publicly known algorithms to veRify the mail system agent's signature

Notations of the form:

UAO->MSAP: procedure statement

designate a dialogue between two functional entities of the mail environment's model, and notations of the form:

MSAP: statement(s)

designate internal operations within a functional entity.

It is assumed further that the mail system agent performing the posting dialogue implicitly knows the originator's slot identifier and that the mail system agent performing the delivery dialogue implicitly knows the recipients slot identifier.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

ACTIVE MESSAGE PROCESSING: MESSAGES AS MESSENGERS

John Vittal

Bolt Beranek and Newman Inc.
50 Moulton St.
Cambridge, MA 02238

Messaging systems have been in active use for many years, using text as the primary means of information transfer. Messages, though, can be active entities which can do much processing on their own. This paper explores the notion of active messages and describes a system which implements them.

BACKGROUND AND GOALS

Electronic mail systems have existed for many years. At the simplest level, electronic mail systems transmit messages (a unit of communication) between originators and recipients. Often, a computer mediates the process, with the resultant system being called a computer-based message system (CBMS). Computers serve both to mediate the actual communications between systems and to provide users with facilities for reading and creating messages.

Today's electronic mail systems are mostly text-oriented. However, one can imagine messages containing information not represented as text, such as graphics, facsimile, digitally encoded voice, and procedures. Most of the research and development efforts in the past eight years have been centered around treating messages as passive information: either pure text or facsimile. One effort, in fact, is building a system which combines text and facsimile [32]. Some current research and development efforts are concerned with what other media, such as graphics and voice (e.g. at IBM), can also be included in messages, and what kind of hardware and support tools are required to allow users to create and manipulate these advanced forms of messages.

However, multi-media messages containing voice and graphics are not the only kind of capability that an advanced message system could be concerned with. This is illustrated by the fact that, quite often, users of computer-based message systems (CBMSs) express a desire to have several very general and dynamic facilities integrated with the more traditional message processing functions. These facilities are:

- A mechanism which resides between a user and his incoming mail, and performs some amount of preliminary processing of his new mail. Some types of processing which might be useful are: "routine" sorting into topics, automatic redistribution of messages about specific topics, automatic

replies and return receipts, and automatic notices such as a message indicating that the user won't be answering mail until he returns from vacation on 31 March. In general, this mechanism provides a way for users to give "standing orders".

- Active message capabilities (i.e. messages as executable objects or procedures) which allow messages to perform some actions on their own. These capabilities would allow the sender of a message to specify his intentions about what can be done with that message, how it should look and behave when a recipient asks to see it.

- A modifiable command structure and user interface which allows users to add their own functionality to the system, create their own interfaces, or modify existing functionality or interfaces. For example, a user might want to make the "Reply" command in a message system automatically move the message into a special file as soon as the message had been answered. Another is to change the size of some of the default windows in a display-based user interface. A third might be to change the system mechanism of command invocation from menus to typed commands. All of these are possible today, and indeed some exist in some systems. What is lacking is a mechanism to allow the expression of functionality in a general way.

Some of these goals are the same as much of the distributed systems work that has been proceeding for the last ten years or so, such as the U. C. Irvine Distributed Computer System [15, 20, 16].

In the Fall of 1977, we started exploring some issues based on the above ideas, primarily the second. Without constraining ourselves to work within an existing mail system environment, we built a preliminary demonstration of capabilities, some of which are shown later in this paper. Significant inroads were made toward solving some research issues associated with processing active messages; however, contract time limitations precluded further development.

We had four primary goals which directed the research:

1. to explore "active" messages,

2. to develop a message-oriented language, through which application implementors could specify specialized functions and automated procedures,

3. to provide a rich, flexible working environment for the creation of the facilities necessary for complex message processing and to support the above two goals, and

4. to provide a flexible command structure and user interface to the functionality provided by the first three goals.

Although most of the work focused on satisfying the first three goals, two separate user interfaces were built.

We created a system which we called a Research-to-Development-Tool for Message Processing (or, R2D2). "Active" messages and a special message oriented language are the implementation mechanism used to

satisfy these goals. The system can send processing specifications in messages. A recipient could then have these "instructions" automatically extracted and executed. Thus, a sender of a message can potentially cause extremely complex activity to take place in a recipient's CBMS environment. Users could also define their own commands or modify system defined commands and functionality.

Much of this work is directly applicable to systems for office automation. Office automation tools are being introduced to organizations, often in the form of message processing systems. Our endeavor to provide a process description language for distributed office processing is not the only one: the process description language which was to be part of Xerox PARC's OFFICETALK system [14] had many of the same goals. IBM, the Wharton School at the University of Pennsylvania, and others are also working on this problem. However, in this paper, we do not go into the details of the relationship between active messages and general office tools and environments, except to say that they can be an implementation mechanism for many office automation functions.

The rest of this paper is concerned with an explanation of active messages, and the particular realization of them in the R2D2 system. This includes a description of the message processing language that was developed and the user interfaces which were built. Some examples of how it was used are given, and the conclusions and open research issues are discussed.

ACTIVE MESSAGES

The simplest characterization of an active message is a "message with a mission that the message knows about". The message must therefore contain (or be) a procedure (i.e. a set of instructions) which is activated sometime after the receipt of the message at its destination host; this could occur sometime before delivery to its actual recipient. Active messages must understand what processing can be performed on them and not allow certain process to be performed. In addition, an active message is potentially self-modifying; a message can change itself in some way at an intermediate destination before it continues on its mission.

These ideas are not unique to message processing. They have existed in computer science in various forms for several years, primarily in the discipline of programming languages. Languages such as Lisp assume that data is program and executable; programs can create other programs and execute them. Object-oriented programming languages such as Simula [10, 11] and Smalltalk [19] have the notion that particular instances of a data type (called a class) know how to perform operations on data of that type. For example, it is the responsibility of an instance of an integer to understand the "+" operator and therefore the addition of integers, rather than the "+" operator understanding how to do the coercion of its arguments and the actual performing of the addition operation.

To show what can be meant by active messages, lets consider three separate examples: a questionnaire, an appointment scheduler, and a purchase order system. Each is of increasing complexity. For each example we give a simplified cursory description and then list the requirements for a system which implements the example. These examples will indicate how active messages can be used to solve some

complex interactive procedures. In addition, the examples provide insight into the nature of a system that embodies active messages. All are examples of tools which, someday, many hope to have in commercially available office automation systems.

Questionnaires

A simple questionnaire asks a series of questions and gathers responses to those questions. After all responses have been received (usually determined by a cutoff date), the results are tallied. In the paper world, respondents specify the answers on the same piece of paper which asks the questions. Branches are handled by statements such as "If the answer to question 5 is 'NO', please skip to question 13".

Typical questionnaires using traditional message systems are similar to questionnaires in the paper world. They are either very short, requiring a small number of answers, or they must be treated as paper questionnaires because of the complexity of the answer format. Complicated questionnaires require respondents to obtain a hard copy of the message which contains the questionnaire. Then, the respondents must create a message consisting of the answers. Note that there is no formal way for the originator of the questionnaire to specify the form of the answers to the questions for his tallying process. Tallies to both of the above generally require some human intervention and cannot be automatically calculated.

An "active" questionnaire can behave very differently. The act of displaying on a terminal a message which is a questionnaire could, rather than just list the questions, ask a question, obtain a response, and based on the response ask more questions. As answers are provided, a message replying to the questionnaire is being automatically composed. When all questions have been answered, the reply is sent to the originator of the questionnaire.

Tallying the answers can be automated to some extent. The replies could contain instructions that indicate what to do with each reply to the questionnaire and what to do with all the answers. For example, each reply could be placed in a particular file; if the file contains a certain number of messages or if a certain amount of time has elapsed since the questionnaire was sent, a tallying process could be invoked on the answers residing in the file. These instructions could have been part of the original message and automatically inserted into the reply.

To enable this facility to be embedded in a message system, the system needs:

- A language to specify the structure and substance of the questionnaire.
- A mechanism to send the instructions in a message and have them evaluated by a recipient's CBMS.
- A mechanism for invoking these instructions when the message embodying the questionnaire is typed, along with a capability to act differently depending on the output device.
- A mechanism to send the replies back to the originator.

Note that a recipient, on acceptance of a message, implicitly accepts all of the processes implied by the message. In the questionnaire example above, this could include an action like:

> reply to the questionnaire in two days or send the originator of the questionnaire a message

These examples indicate clearly that the originator of a message is allowed access to everything that happens to the message, that the message is a live object with intermediate stops, and that it is self-modifying.

Current production message systems do not have these capabilities. These capabilities, and those described below, are a necessary component of the truly automated office system.

Calendar management

Everybody knows something about calendars and scheduling appointments. The following activities are useful in calendar management:

- Schedule appointments and meetings, often with several individuals and much interaction to determine a time when all can participate
- Keep track of the schedule
- Show the current state of a calendar for any time period
- Show what appointments are currently being negotiated
- Provide appropriate notification of an impending appointment

It would be beneficial if computers could mediate the interactions required for scheduling in a very active and automatic manner. In order to do this with a CBMS with active messaging capabilities, the following facilities are required in addition to those needed for questionnaires:

- A daemon process to manipulate mail before it is actually delivered to the user. The daemon may decide to notify the user in some way of the arrival of a calendar schedule message.
- A data base of the current state of the calendar. This could be something as simple as a data base of messages or a more complex specialized data structure.
- An ability to query remote databases and take knowledgeable actions. These actions are replies to requests like "when are you available for a meeting before Friday?," or "schedule an appointment with me at 3 p.m. tomorrow."

Active messages are the mechanism for querying the data bases and taking knowledgeable action. With active messages and these scheduling specific facilities, it is possible to create an

automatic calendar management facility which requires little human intervention by any other than the person requesting the meeting. Automatic scheduling, however, is a very hard problem to solve generally. These facilities will not solve the problem. Rather, they provide a framework for the development of facilities like interactive calendar management and a mechanism for the future development of automatic systems.

Purchase orders

Purchase orders are even more complicated. First, a PO needs to be filled out. Then it must be approved, possibly by several levels of management, before it finally ends up in the purchasing department's hands. Then the item must be procured and delivered to its requestor.

There are organizational policies that govern the number of signatures required based on the kind of purchase being requested and its value. Some organizations have the policy that no purchase for over $100,000 can be made without the express approval of the corporation president. Most individuals don't know what the corporate policy is, but the system could. If you ask to purchase some items totalling $100,100, the system knows to whom the PO should be routed for signature. Also, the system could inform you that if you cut $100 from the PO, the request could go out sooner since the president will not be in the office for a week.

Requestors also like to be able to query the state of the PO at any time. How often has the question "What's the state of my order for ten widgets?" been asked?

This example requires some additional facilities over those required for the questionnaire and calendar management examples:

- A knowledge base consisting of a representation of organizational policy as it relates to purchase orders.
- Memory. This enables future requests about the state of a PO.
- Situation recognition. A way to notice when one of the policy conditions occurs.

Situation recognition is a very hard problem in the computer science domain of Artificial Intelligence, and currently remains unsolved. The generalized purchase order problem and those like it are much further down the road than the others noted here.

Again, active messages provide a framework for automating procedures to some extent by allowing users or organizations to specify what the actual procedures are. This allows interim solutions to the general problem. As solutions to problems such as situation recognition are developed, they can easily be included into the framework set up by active message processing.

SOME CAPABILITIES OF THE CURRENT IMPLEMENTATION

The system was built incrementally, starting from a kernel on which certain basic capabilities were implemented, expanding the kernel as

sers of the system discovered new needs. This iterative development process pointed out several new capabilities that were added to the various languages. Some of these new capabilities were fairly simple. One allowed for the presentation of a question and gathering of a response which had to fit into a set of possible answers. Another was the ability to do string matching on the contents of a field of a message. Yet another was the ability to do standard numerical operations such as obtaining the length of a set of messages, numerical comparisons, and arithmetic. The general capabilities of R2D2 are described in the section on "The Message Processing System".

More complex capabilities were also added; one is a rudimentary questionnaire facility. A user who wishes to send a questionnaire prepares a special message and transmits it to a set of respondents. The respondents' message systems are able to interpret the instructions in the message. In the case of a questionnaire, the instructions prompt the user through a series of questions, and compose a draft message containing the respondent's answers to those questions. This draft is then transmitted back to the originator of the questionnaire. A sample interaction with such a questionnaire is shown in Figure 1, with user-input underlined. A portion of the message created by the last of the user-interactions in that figure is shown in Figure 2.

It is interesting to note that the facilities required to implement a simple, personal, non-automatic calendar management facility are not very different from those required to implement a questionnaire. To schedule an appointment, a message asking a question like

> When this week will you be available to meet with Joe, Dan, and myself to discuss the design for FCA?

can be sent and an answer automatically obtained and returned. It may take several iterations to reach a time convenient to all. At that time, an announcement of the meeting time and location can be distributed to all intended attendees. A daemon process for announcing impending appointments may be useful.

A second complex capability was that of message circulation, the serial distribution of a message to its recipients. It is done by including a distribution list as part of the message; when the message is first sent, it is sent to the first recipient on the distribution list. The recipient, or someone the recipient delegates, sends the message on to the second recipient on the list, perhaps after commenting on or adding to the message. This continues until all recipients on the distribution list have received the message. Each subsequent recipient was able to see the comments added to the message by all of the previous recipients of the message. The comments were output side-by-side in columns, one column per recipient comment, unjustified. Figure 3 gives an example.

THE MESSAGE PROCESSING SYSTEM, R2D2

R2D2 is an experimental system. Its main goal was to develop a mail processing environment which is much more dynamic than typical current systems by allowing:

```
< PRINT 3 CD
This is a questionnaire about baseball.
Do you want to answer it now? No
Thank you - please remember to respond to this questionnaire
  before Thursday.

< PRINT 3 CD
This is a questionnaire about baseball.
Do you want to answer it now? Yes
Do you follow baseball? No
Thank you very much...

< PRINT 3 CD
This is a questionnaire about baseball.
Do you want to answer it now? Yes
Do you follow baseball? Yes
Which team do you root for? DODGERS
How do you think the DODGERS will do in the pennant race?  ?
one of:
1st
2nd
First
Last
Other
Second
Uncertain

How do you think the DODGERS will do in the pennant race? First
Will they win the World Series? Yes
Thank you very much...

<
```

Figure 1
Questionnaire examples

- o messages to be active, executable objects
- o senders to specify the actions that a message can take or that users can perform on the message after it has been sent
- o users to tailor the functions that their system provides, in addition to adding new functionality
- o users to tailor the interaction style of the system

The R2D2 system is based on the current TENEX/ARPANet paradigm for mail posting, transport and delivery [9, 31, 33]; that is, messages are only textual in nature. This choice was made for ease in experimentation. It permitted us to experiment with the important issues of functionality without having to build new mechanisms for

```
Date: ...
Subject: Response to Baseball Questionnaire
Instructions: ...
To:  VITTAL
Q1: YES
Q2: DODGERS
Q3: 1ST
Q4: YES
```

Figure 2
Example message generated by questionnaire

message posting, transport, or delivery. However, neither the design of the system or the message processing environment we created is dependent on this choice. Although the tools and the language did not support forms of data other than text, the basic framework was designed and exists to easily include them. This paradigm was sufficiently flexible to permit us to experiment with the sending, and therefore the subsequent execution of instructions in messages, as has been shown by the examples.

R2D2 consists of three major parts:

- Messages. These are either messages which have been received or those which are being prepared to be sent.
- Instructions. There are three kinds of instructions which make up the message processing language used to implement active messages:
  - instructions for selecting subsets of messages (Selectors)
  - instructions for transcribing messages[2] (PInstrs)
  - instructions for composing messages (CInstrs)
- User-interface. The user interface consists of two parts:
  - commands that users can invoke to perform some operations
  - a presentation mechanism

In order to determine the "right" primitives for such a message processing language, rather than build just one language, we built several - one for each kind of instruction. There are few commonalities between the languages. In the future, to provide a more uniform and consistent language, the three parts should be integrated into a single language.

The message as Jill receives it. It has already been circulated to Elisa Beth and Dokes.

```
Date: ...
Subject: Please comment and circulate
Instructions: ...
Originator: Sampson
From: Dokes
To: Jill
Circulate-To: Elisa Beth, Dokes, Jill, Sampson
Circulate-Next: Sampson
Comments-Elisa Beth: Fine by me!  When is it going out?
Comments-Dokes: I think the second paragraph should be much
 stronger.  It doesn't reflect the attitude of the group.
Text: The following is the text of a message I want to send
 to Joe Dawes.  What do you think?

 ...
```

When Jill prints the message, she sees the following on her screen:

```
< PRINT 12 CD
Jill, Please comment and circulate

The following is the text of a message....

Here are comments:

      Elisa Beth                    Dokes

      Fine by me!  When is it       I think the second paragraph
      going out?                    should be much stronger.
                                    It doesn't reflect the
                                    attitude of the group.

Do you want to comment now?  Yes
Type your comments; end with ^Z:
Ok.  I think Joe is crazy, but he always disagrees.^Z

<
```

Figure 3
Message circulation example

A user interface to these objects and the message system embodying them are completely separate in R2D2. We were not as concerned about the user interface as we were about the functionality of a powerful message system. But, in order to try out some of the ideas embodied by the functionality, a somewhat generalized interface was

built.

Some message systems have allowed users to modify the behavior of the system in fairly specific ways that were foreseen by their developers. Hermes [29] and MS [8] are two such systems. However, the notion of allowing users to provide general specifications for complex behavior is new to message processing even though it has been used in other aspects of computer software for several years. This idea has been available in current or developing user tools, most notably in recent years with text preparation systems such as PUB [38], Scribe [36], and R [39], and with text editors such as Teco [35] and Emacs [7, 41]. They have also been available in tools for programmers such as the programming environments provided by Interlisp [27, 43], DLISP [42], and various MacLisp dialects [26, 47], in addition to executive-level tools such as those provided by MXEC [1].

The rest of this section discusses the particular considerations that went into the development of the instructions and the user-interface in R2D2. Included is a description of the language for instructions and a description of the mechanism for realizing active message processing.

Considerations for instructions

In R2D2, transcription and composition instructions are regarded as sets of instructions whose execution will result in a message being transcribed (most often, printed) or created (composed). Selectors enumerate sets of messages. These instructions and some simple associated mechanisms allows two separate styles of usage of the system. One permits users to program their own message processing applications or modify system-defined functionality. The other permits the realization of active message processing.

Some of the operations that transcription instructions need to include are formats for output (including a mechanism for columnar printing of text), control structures, testing for field existence or contents, indirection (like calling a subroutine by having one PInstr be able to invoke another), and the ability to "call" a composition instruction (that is, a PInstr can invoke a CInstr and vice versa)[3]. In addition, instructions can be extracted from the message being transcribed to be used as further printing instructions or for inclusion into the message processing environment.

Formatting operations for message processing are, in some sense, a superset of those desired for text processing. Not only do all of the standard document preparation operations need to be available and easily dependent on the type of printing device, but there also needs to be a printing specification mechanism for constructs "higher" than just text, such as message fields which contain such information as the field name, comments on the field, and so on. One of the requirements is the ability to obtain the contents of fields of messages, or obtain other information about the message which is not specified explicitly in the message itself (such as status information, which file the message resides in, etc.). Further, one needs to be able to specify in what order, where and in what format this information is to be printed.

The mechanism used to send instructions (that is, PInstrs) in a

message and have those instructions interpreted must be a standardized one. This mechanism need not be complex. Messages in R2D2 are printed (transcribed) using a default PInstr which looks for the existence of an "Instructions" field. If that field exists, then its contents are used to transcribe the message, otherwise, the instructions from the receiver's environment are used. Notice that this mechanism can not only transcribe a message through instructions obtained from the message itself, but it can also cause composition to occur.

The instructions which deal with message composition are also very rich. The language includes mechanisms to allow for replacement, deletion or appending to the contents of fields of the message being created. Information to be added to the draft message being created can come from the user, a literal string, or other messages under both program and user control. Also, various control structures are included in the language to permit iteration over those existing messages. This is because some acts of composing (for example, the typical operations of forwarding or replying) use an initial set of existing messages in the creation of a new draft message. Using these facilities allows simple behavior such as the standard message processing functions Reply and Forward to be easily implemented using composition instructions.

When composing a message, one must have the ability to specify prompting information, the location of the prompts on the screen, the fact that data has to be gathered, the type of the data (such as a single line of text, a body of text, a date, an ordered combination of data types, and so on), and perhaps the constraints the data must adhere to. As with transcription instructions, textual data is the only allowable kind. The framework for other types of data, such as graphics or facsimile, is included, but no facilities were developed to allow for the creation, editing, and transmission of data other than text. The use of a structured format for message transmission, such as that being proposed as a standard in the U.S. by its National Bureau of Standards [12], make dealing with multi-types of data much easier.

Due to several considerations, the emphasis on the development was placed on generalized facilities in the language and not on the formats for input. The prompting mechanism was strictly sequential -- print some data, give a prompt, read some input, and so forth. The questions of presentation mechanisms were interesting user interface questions which required an interface to the CInstr language, but we did not demonstrate these aspects.

Selectors are very different from PInstrs and CInstrs. Rather than manipulating the contents of messages, Selectors create sets of messages by examining the contents of messages and using the standard set operations. Selectors are used to provide arguments to the transcription and composition processes.

The instructions and mechanisms described above provide users with a very powerful tool. They take message processing out of the hands of the system developers and put the power in the hands of a set of users. Mail creators can cause extremely complex actions in mail recipient's environments; some instructions could completely destroy a user's environment in an unrecoverable way. Restraint must be exercised prior to releasing a production version of this facility to ensure that proper access control mechanisms are included in the

system to prevent indiscriminate access and modification of a message recipient's user environment. We did this in R2D2 by "trapping" the mail sending processor and the object creation and deletion process, when the processes were invoked by instructions in a message, in order to force user confirmation and let the user know what was happening "behind his back".

Considerations in the R2D2 User-Interface

The user-interface to R2D2 consists of two separate parts: user-level command definition and presentation style. Commands are the way that users direct system operation. Presentation style is not only how commands are invoked, but how data appears on the screen. It can make use of both standard character-oriented terminals and graphics terminals.

The intent of providing a flexible and expandable command definition language is to amplify the scope of user-level operations that can be performed with the system. One very useful capability would be to enable users to define command procedures or macros which can be used to automate certain tasks. Users of R2D2 are allowed to specify their own commands. System definitions of a command can be overridden. Command definitions can use other system or user defined commands in its definition. We did not define a special language for defining commands; the language used was the implementation language, Interlisp.

Another capability would be to provide instructions to a daemon process which can execute "background" tasks, initiate and monitor batch-like tasks, and provide notice of events of interest. This was not implemented in R2D2.

These capabilities, and many others, have been understood for quite some time, having been implemented in other contexts such as multiple connection TELNET, RSEXEC, Unix pipes, Multics execcom, and so forth. They have recently been integrated at BBN into a coherent Exec-level package called MXEC [1] available under the DECSYSTEM-20 operating system. Part of the R2D2 research effort was to provide a message-oriented command definition mechanism as part of the system; other aspects of flexible functionality mechanisms were beyond the scope of this research.

In developing R2D2, we intended to provide a flexible environment and a tool for the development of various presentation mechanisms to a single system. The same precepts used in the design of the R2D2 system were used in the design of the user-interface system. However, the questions which we hoped to answer were vastly different. Specifically, we were trying to understand if a generalized mechanism could be developed to allow terminals of varying capabilities to interface in interesting terminal dependent ways to the underlying system.

To this end, we built two user interfaces. One is a standard full-duplex interface of the style used by several systems such as Hermes and TENEX [4, 28]. These systems require users to specify commands a line at a time; they were designed for teletype-like terminals and do not use many of the features provided by current cursor-addressable terminals or bit-mapped terminals. The other user interface uses a bit-mapped terminal built by BBN which had a windowing capability and variable-pitch fonts.[4] The graphics-

oriented terminal system [21] is designed for area-type manipulations, much in the spirit of Sproull [40] and Teitelman [42]. This allowed a presentation mechanism which was more screen-oriented and took advantage of the graphical capabilities of the terminal.

These two interfaces did not give us sufficient experience to make many generalizations about the nature of a terminal-independent presentation mechanism. However, the aspects of several human factors were noted as candidates for inclusion in a data presentation language. Several other systems have come to similar conclusions:

- Two-dimensional message prompting and composition (i.e. form filling) is important if the terminal can support it. That is, allowing for displaying a set of fields (prompts) on the user's terminal, and allowing the user to fill in the fields in any order desired. This could interact negatively with the standard sequential prompting and acquisition mechanisms when strictly sequential processing is required, as in the questionnaire example.

- Primitives for driving terminals must be provided, including dynamic cursor positioning, blanking of small areas of the screen, and so forth. This interacts with the user interface mechanisms such as windowing and scrolling.

- Fonts are useful for highlighting, printing of text, and readability. The language, however, needs to be aware of variable-pitch fonts when formatting data.

- The size of the screen is important when considering windowing capabilities.

- Errors or help can overlay other parts of the screen. There does not have to be a specific area dedicated to error messages or on-line assistance.

- A special area of the screen for dynamic notices is necessary. These are notices that a daemon mechanism might present to the user when a certain condition has been met.

CONCLUSIONS AND FURTHER RESEARCH

R2D2 is a message system where messages are active entities rather than being just passive communication objects. It allows a sender to specify the actions that a message can take upon its receipt. In addition, R2D2 provides a highly flexible user interface, allowing tailoring of both the functionality and interaction style of the system.

The work reported here is only a start on a large set of issues which may or may not be directly related to messages and message processing. We defined many of the primitives in a message-oriented language used to implement active messages and looked at some of the operations and styles which are useful to have in a rich computer-based message system. Our experience indicates that these capabilities would be very well received by users of production-level message systems. Allowing users access to the low-level

system functionality and permitting users to modify and manipulate that functionality is something that many users have been requesting for a long time.

The examples given previously point out that active messages can exist, are live objects that make intermediate stops, and can modify themselves. However, before the ideas enumerated in this paper can be included in a system for widespread use, several additional research areas must be explored.

First, the actual language available for user-specification of capabilities needs to be explored. As already mentioned, the work we did was on the primitives that such a language must have, but not on the actual form of the language. The primitives of the language have not been fully specified, as is obvious by some of the other research areas mentioned below.

A second research area is that of a command specification language for specifying new commands and macros and interfacing these to other system and user-defined commands.

Third, an "interface" between the language(s) and the user's hardware. Some aspects have been handled in the languages, such as the inclusion of fonts and testing primitives for the type of terminal being used. However, the languages must be expanded to include display handling primitives, such as specification for windows on the screen, and manipulation primitives for those objects. This will lead to a language for specifying interaction style which will need interfacing to all the other languages.

Fourth, an editor for the language(s) is necessary. Experience has shown that editors tuned to the syntax and semantics of a language are more useful than a character-oriented editor. However, the human factors for a system having two radically different editors, one for text and another for instructions, must be examined and the results fed back into the design of the ultimate system.

Fifth, a set of tools to permit the checking of instructions in a protected environment needs to be provided. Often, instructions can be very complicated and users can get themselves into a great deal of trouble. User's should be able to make sure what will happen if some instructions are executed. There should be a mechanism which protects the user from doing anything disastrous to his environment. Yet, it should allow him to operate with impunity on real data. This would save the effort of constructing the best data structures, and if the instructions work as planned, should allow the user to indicate that the work just performed can be taken to have been actual work rather than as just protected "debugging".

Finally, access controls should be provided to allow users to specify what operations can be done in his environment without being notified, which he should be notified about, and which are not allowed. This is to provide protection against externally generated instructions performing actions in a local environment against the wishes of the local user. These access controls should be applied when the instructions are executed. They should also be determinable by organizational policy.

Along with access controls, however, is the potential for a more complex task: that of inspecting an instruction before it is

executed to see what it will do, and to check if it might violate some of the user's access privileges.

Another kind of access control on messages is an indication of what kind of processing is allowed on the message itself. For example, an originator might not want the message forwarded to anybody, a few specific people, or not to a few specific people. Being able to restrict recipient processing at this level is very powerful.

A broader issue than those addressed above concerns who is capable of using such a system. Those who are highly experienced with computers or are very technically savvy are used to using systems of such complexity. But, can non-programmers use such a system? Even non-technical support personnel? These are questions which must be answered prior to a large-scale public offering of such a system. It could mean some amount of organizational restructuring.

The above research areas are not only message oriented, but apply to much of computer science. Nor are they easily solved. However, we believe that the results of research into these areas will provide a positive impact on users of message systems and other office systems, not to mention the systems themselves.

ACKNOWLEDGEMENTS

This work was supported by the Defense Advanced Research Projects Agency of the US Department of Defense, Information Processing Techniques Office.

I would like to thank Ted Myer, project leader, for support in this research. In addition, special thanks go to Ron Brachman and D. Austin Henderson for the many discussions about the work and help with the implementation.

Mario Grignetti and Dave Crocker deserve special recognition for providing much assistance through their extensive comments on early drafts of this paper.

FOOTNOTES

[1]This knowledge might never be officially tolerated even though it is common practice by employees to provide services such as this to circumvent corporate procedures. The mechanisms may be provided in a system for actions such as this, but their use in this way is a matter or corporate policy.

[2]The notion of transcribing is more general than just printing. Messages may be transcribed for inclusion into another message rather than just formatted for printing on a terminal or other output device.

[3]The notion of printing a message could be considered nothing more than a complex mechanism for prompting users for input during the composition process.

[4]The effects of the bit-map interface could be simulated on many of the currently available cursor-addressable scope terminals, such as the Human Designed Systems, Inc. Concept-100 or the Digital VT-100.

However, the size of those screens are generally too small to allow for the number of windows of reasonable size that we provided using the BBN bit-map. They also do not have a flexible multi-font capability.

REFERENCES

[1] W. L. Ash.
MXEC Reference Manual.
Technical Report 4372, Bolt Beranek and Newman Inc., Cambridge, MA, February, 1980.

[2] N. Bergbach and H. Grosser.
Text Communications by Wire Illustrated with Reference to TELEX.
Telecom Report 1(1):19-21, August, 1978.

[3] M. S. Broos, S. W. Galley, and A. Vezza.
Data-based Message Service - User's Manual
MIT, LCS, Cambridge, MA, 1978.

[4] J. D. Burchfiel, E. M. Leavitt, S. Shapiro, and T. R. Strollo.
TENEX User's Guide
3rd edition, Bolt Beranek and Newman Inc., Cambridge, MA, 1975.

[5] L. J. Cartalano.
A Sub-Minute Digital Facsimile Machine, Description and Applications.
In Proceedings of the International Conference on Communications. IEEE, New York, 1978.
Toronto, June 4-7, 1978.

[6] V. G. Cerf.
The Electronic Mailbox, A New Communication Tool for the Hearing Impaired.
American Annals of the Deaf 123(6):768-72, October, 1978.

[7] E. Ciccarelli.
An Introduction to the EMACS Editor.
AI Memo 447, MIT AI Lab, January, 1978.

[8] D. H. Crocker.
Framework and Functions of the 'MS' Personal Message System.
Technical Report R-2134-ARPA, The Rand Corp., 1977.

[9] D. H. Crocker, J. J. Vittal, K. T. Pogran, D. A. Henderson, Jr.
Standard for the Format of ARPA Network Text Messages.
RFC 733, The Rand Corporation, Bolt Beranek and Newman Inc. and MIT, November, 1977.

[10] O-J. Dahl and K. Nygaard.
SIMULA -- An Algol-Based Simulation Language.
CACM 9(9):671-678, September, 1966.

[11] O-J. Dahl, B. Myhrhaug, and K. Nygaard.
SIMULA -- Common Base Language
Norwegian Computing Center, Oslo, Norway, 1970.

[12] D. Deutsch, R. Resnick and J. Vittal.
Specification of a Draft Message Format Standard.
Technical Report 4486, Bolt Beranek and Newman Inc., November, 1980.
Also NBS Report No. ICST/CBOS - 80-2.

[13] C. A. Ellis and G. J. Nutt.
Computer Science and Office Information Systems.
Technical Report SSL-79-6, XEROX Palo Alto Research Center, 1979.

[14] C. A. Ellis and G. J. Nutt.
Office Information Systems and Computer Science.
Computing Surveys 12(1):27-60, March, 1980.

[15] D. Farber, J. Feldman, et. al.
The Distributed Computing System.
In Proceedings of the Seventh Annual IEEE Computer Society International Conference, pages 31-34. IEEE, February, 1973.

[16] D. J. Farber.
Software Consideration in a Distributed Architecture.
IEEE Computer, March, 1974.

[17] J. D. Foulkes.
The Application of Contextual Analysis in Message by Optical Character Recognition.
Technical Report, Arthur D. Little, Cambridge, MA, 1976.
Available as NTIS PB-260 776/OST.

[18] J. D. Foulkes.
A Survey of Factors Affecting the Quality of Data Compressed Black/White Facsimile.
Technical Report, Arthur D. Little, Cambridge, MA, 1976.
Available as NTIS PB-260 777/OST.

[19] A. Goldberg, A. Kay, editors.
Smalltalk-72 Instruction Manual.
SSL 76-6, XEROX Palo Alto Research Center, Palo Alto, CA, March, 1976.

[20] E. Gord and M. Hopwood.
Nonhierarchical Process Structure in a Decentralized Computing Environment.
Technical Report 32, Department of Information and Computer Science, University of California, Irvine, CA, June, 1973.

[21] N. Greenfeld, F. Zdybel, E. Ciccarelli, and M. Yonke.
BMG Reference Manual.
Technical Report 4368, Bolt Beranek and Newman Inc., April, 1980.

[22] J. Heafner, L. Miller, B. Zogby.
Sigma Message Service Reference Manual.
Technical Report ISI/WP-5, Information Sciences Institute, February, 1977.
Working Paper.

[23] D. A. Henderson and T. H. Myer.
Issues in Message Technology.
In Proceedings of the Fifth ACM/IEEE Data Communications Symposium, September 27-29, 1977; Snowbird, UT, pages 6/1-9. IEEE, New York, 1977.

[24] H. G. Lehtman.
Toward Adaptive Man-Computer Interfaces.
Technical Report, Stanford University, Dept. of Industrial Engineering, Stanford, CA, 1977.

[25] R. Levin and M. D. Schroeder.
Transport of Electronic Messages through a Network.
Technical Report CSL-79-4, XEROX Palo Alto Research Center, Palo Alto, CA, 1979.

[26] D. A. Moon.
MACLISP Reference Manual
MIT, 1974.

[27] J. Strother Moore.
The Interlisp Virtual Machine Specification.
CSL-76-5, Xerox Palo Alto Research Center, Palo Alto, CA, September, 1976.

[28] T. H. Myer and J. R. Barnaby.
TENEX Executive Language Manual for Users
Bolt Beranek and Newman Inc., Cambridge, MA, 1971.
Revised April, 1973 by W. W. Plummer.

[29] T. H. Myer and C. D. Mooers.
HERMES User's Guide
Bolt Beranek and Newman Inc., Cambridge, MA, 1976.

[30] T. H. Myer and D. W. Dodds.
Notes on the Development of Message Technology.
In Proceedings of the Berkeley Workshop on Distributed Data Management and Computer Networks; May 25-6, 1976, pages LBL-5315:144-154. University of California, Lawrence Berkeley Lab., Berkeley, 1976.

[31] T. H. Myer and J. J. Vittal.
Message Technology in the ARPANET.
In NTC '77. IEEE, December, 1977.

[32] T. H. Myer.
Personal communication.

[33] N. Neigus.
File Transfer Protocol.
In ARPANET Protocol Handbook, pages 265-298. Network Information Center, January, 1978.
Originally published as RFC 542, August 12, 1973.

[34] S. S. Owicki.
Specification and Verification of a Network Mail System.
Technical Report CSL TR-159, Stanford University, Computer Systems Lab., Stanford, CA, 1979.

[35] W. Plummer.
TENEX TECO: Text Editor and Corrector Manual
Bolt Beranek and Newman Inc., Cambridge, MA, 1973.

[36] B. K. Reid and J. H. Walker.
SCRIBE Introductory User's Manual
Third edition, CMU and Bolt Beranek and Newman Inc., 1980.
Published by Unilogic, Ltd.

[37] R. L. Rivest, A. Shamir and L. Adelmen.
A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.
ACM Communications 21(2):120-126, February, 1978.

[38] N. Smith.
Sumex-AIM PUB Manual
SUMEX, Stanford, CA, 1975.

[39] A. Snyder and E. Moss.
R Reference Manual
25 June 1979.

[40] R. F. Sproull.
Raster Graphics for Interactive Programming Environments.
Computer Graphics 13(2):83-93, August, 1979.

[41] R. M. Stallman.
EMACS: The Extensible, Customizable, Self-Documenting Display Editor.
A.I. Memo 519, MIT AI Lab, June, 1979.

[42] W. Teitelman.
A Display Oriented Programmer's Assistant.
In Proceedings of the Fifth Internaitonal Joint Conference on Artifical Intelligence -- 1977, pages 905-915.
International Joint Conference on Artificial Intelligence, August, 1977.

[43] W. Teitelman, et. al.
Interlisp Reference Manual.
Xerox, 1978.

[44] M. Turoff and S. R. Hiltz.
Development and Field Testing of an Electronic Information Exchange System.
Technical Report 9, New Jersey Institute of Technology, Computerized Conferencing and Communications Center, Newark, 1977.

[45] J. J. Vittal.
MSG Manual
Bolt Beranek and Newman Inc., Cambridge, MA, 1976.

[46] D. A. Waterman.
A Ruse-Based Approach to Knowledge Acquisition for Man-Machine Interface Programs.
Rand Paper P-5823, The Rand Corp., Santa Monica, CA, 1978.

[47] D. Weinreb and D. Moon.
Lisp Machine Manual
MIT AI Lab, 1979.
Second Preliminary Version.

COMPUTER MESSAGE SYSTEM PROTOCOLS

Chairman - N. Naffah

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

DESIGN OF A MESSAGE FORMAT STANDARD

Debra Deutsch
Bolt Beranek and Newman Inc.
10 Moulton Street
Cambridge, Massachusetts 02238

A draft standard for the format of messages produced by computer based message systems (CBMSs) has recently been developed. The draft standard is discussed in terms of its relationship to other standardization efforts and its design rationale. An overview of the draft standard is provided. Important design issues are singled out and areas for possible future standardization are suggested.

INTRODUCTION

A computer-based message system (CBMS) allows communication between "entities" (usually people) using computers. Computers serve both to mediate the actual communications between systems and to provide users with facilities for reading and creating the messages.

Computer-based message systems have been developing for over ten years. An early leader in the development of the technology was the Defense Advanced Research Projects Agency of the U.S. Department of Defense. The results of various research efforts, some of which were DARPA sponsored, have been published in the literature. SRI-International has prepared a bibliography summarizing research in computer message services [1].

More recently, CBMSs have been one of the bases in industry for the introduction of office automation. A growing number of organizations use either their own or a commercially available CBMS. The design and complexity of these systems vary widely. Standards are necessary to provide a basis for interaction between different CBMSs. This paper discusses the design and rationale of a draft CBMS message format standard being developed by Bolt Beranek and Newman under contract to the U.S. National Bureau of Standards. A draft version of the NBS Message Format Standard [6] has been published for industry comment and review. This paper reflects the contents of that draft.

The remaining portion of this section provides background for the rest of this paper. A simple functional model for computer based message systems is discussed and some past CBMS standardization efforts are summarized. The next section deals with the draft standard itself. A concluding section suggests some topics and considerations for future work.

A MODEL FOR COMPUTER BASED MESSAGE SYSTEMS

A CBMS permits the transport of a message from an originator to a recipient. "Originator" and "recipient" are used in their normal English senses. A message (in its most abstract definition) is simply what an originator intends to be a unit of communication to a recipient. A CBMS offers several classes of functions to its users:

- Message Creation: The facilities used by a message originator to create messages and specify to whom they are to be sent.
- Message Transport: The facilities used to convey a message to its recipient(s).
- Recipient Processing: The facilities used by a message recipient to process messages that have arrived.

CBMS facilities for message creation, transport, and recipient processing are reflected in a logical model of a CBMS developed by IFIP Working Group 6.5 [12]. (An essentially identical model has been proposed at a CCITT Study Group VII preparatory meeting on question 3C/VII regarding Message Handling Facilities.) It provides both a lexicon and a framework for the discussion of CBMS standards. It also provides a high-level description of both user facilities and system architecture. The model consists of a Message Transport System and a number of User Agents.

A User Agent is a functional entity that acts on behalf of users, assisting with preparing and inspecting messages and communicating with the Message Transport System.

The Message Transport System is an entity that accepts a message from its originator's User Agent and ultimately passes it to each of its recipients' User Agents. The Message Transport System may perform routing and storage functions (among others) in order to accomplish its task.

Transferring a message from an originator's User Agent to the Message Transport System is called Posting; the originator's User Agent and Message Transport System engage in a Posting Protocol in order to accomplish Posting. Transferring a message from the Message Transport System to a recipient's User Agent is called Delivery; the recipient's User Agent and Message Transport System engage in a Delivery Protocol in order to accomplish Delivery.

The point at which responsibility for a message is transferred is called a Slot. The Posting Slot is the point at which responsibility for a message passes from an originator's User Agent to the Message Transport System; the Delivery Slot is the point at which responsibility for a message passes from the Message Transport System to a recipient's User Agent.

The IFIP model further subdivides the Message Transport System into a series of modules called Mail Service Agents (or MSAs) which are discussed in [12].

While a message is inside the Message Transport System it is enclosed in a data structure called an envelope. The envelope

carries information of use to or collected by the Message Transport System, such as the message's final destination, its route (so far) through the Message Transport System, billing information, or service requests. Some of the information on the envelope, such as the message's final destination, may be redundant with information contained in the actual message. Whether the posting User Agent or the posting Mail Service Agent creates the envelope is still under discussion. Likewise, consensus has not yet been reached on how much of the envelope (if any) is passed from the delivering Mail Service Agent to the recipient's User Agent.

ROLE OF PAST CBMS STANDARDIZATION EFFORTS

The draft NBS message Format Standard was influenced by previous CBMS standardization efforts. A major portion of that work has taken place on the ARPAnet, which began its life as a DARPA-sponsored experiment in computer networking in order to provide resource sharing. The development of message systems on the ARPAnet is discussed in a paper by Dodds and Deutsch [2]. It has only been in the past few years that attempts to interconnect CBMSs have spread beyond the research community.

There is one protocol and one standard which currently govern ARPAnet message traffic. The protocol is part of a larger File Transport Protocol [3]. It corresponds to both the Posting and Delivery Protocol portions of the IFIP model, and also specifies the operation of the ARPAnet's Message Transport System. The standard, which is known as RFC 733, specifies the semantics and syntax of ARPAnet messages [7]. In other words, it specifies both the meaning of objects found in messages and how that information may be represented.

The ARPAnet message transport protocol has changed little from its original form. On the other hand, the message format standard has undergone several revisions over the years. Its evolution has paralleled the growing sophistication of the CBMSs themselves. The earliest systems created and read simple, memorandum-like messages. Today there are experimental computer-based message systems which can handle facsimile, graphics, or voice in addition to text. Others may interpret complex instructions found in incoming messages.

The current ARPAnet message format standard reflects the older model of messages as memoranda. It defines a series of message fields which are used for specific purposes, such as identifying who sent a message, when it was sent, its subject and so on. It defines a representation scheme which is wholly based upon the printing ASCII characters plus the characters for space, tab, carriage-return, and line-feed. There are now plans to replace this standard with a new format standard (RFC 753) developed by Postel [8]. RFC 753 retains the use of message fields while abandoning ASCII text in favor of a machine-readable, list-structured format. This type of format is better suited for representing the more complex objects which are beginning to find their way into messages.

THE DRAFT NBS MESSAGE FORMAT STANDARD

In April 1980 work was begun at Bolt Beranek and Newman on a draft

message format standard for eventual adoption by the U.S. National Bureau of Standards as a Federal Information Processing Standard.[1] Its purpose is to promote compatibility between computer-based message systems procured by the U.S. government. The following pages discuss the content and design of the draft standard, as of September 1980. An introductory section describes the scope of the draft standard, how it is organized, inputs used in its design, and the model of a CBMS message on which it is based. The following two sections discuss the two major portions of the draft standard itself: a set of message fields defined by the draft standard, and a standard representation scheme for messages and their components.

INTRODUCTION

This section describes the scope of the draft NBS Message Format Standard, how it is organized, the inputs used in its design, and the model of a CBMS message on which the draft standard is based.

Scope

The draft NBS Message Format Standard specifies the semantics and syntax to be used for messages being exchanged between CBMSs. The draft standard treats message semantics and syntax separately. The semantics section defines a set of fields and their usage. The syntax section defines a representation scheme for data which may be included in messages.

The draft standard does not address the following issues:

- The nature of the Message Transport System, except to specify the assumption that it transports messages transparently.
- The form or nature of the Posting protocol.
- The form or nature of the Delivery protocol.
- Representations for unique identifiers (in particular, message identifiers).
- Network and internetwork addressing [9, 10, 11, 13].
- Representations for identities of message originators and recipients.
- Functions that CBMSs provide for users.
- Representations for multi-media objects.
- Data representation for messages within CBMSs.
- Data sharing or any storage management within CBMSs.
- Representations for fixed or floating point numbers.

The scope of the draft NBS message format standard is essentially the same as that of RFC 733, except that the draft NBS standard does not deal with how message originators or recipients are identified. This was done because naming and addressing is an active research

area. The scope of the draft NBS message format standard is a smaller subset of RFC 753's scope, since RFC 753 deals with the naming and addressing problem and also specifies objects (such as envelopes and mailbags) and commands (such as FORWARD and PROBE) which are associated with the Message Transport System. More experience must be gained with Message Transport Systems before they can be specified in standards used outside the research community.

Design Criteria and Input

The draft NBS message format standard was designed with the following criteria in mind:

1. Viability - the standard should be robust and implementable.

2. Compatibility with existing CBMSs - It should be possible to modify existing systems to comply with the standard without subjecting them to major rewrites.

3. Extensibility - The standard should not be quickly obsoleted by changing technology. Advances in CBMS design, such as multi-media capabilities, should be allowed for without requiring immediate implementation.

The draft standard is based on several documents and the current state of many CBMSs available both in various industries and the research communities. These documents include the standardization efforts in the ARPANet [7, 8], proposed ISO and ANSI header format standards [4, 5], the work of IFIP Working Group 6.5, and various papers about the general nature of mail systems, addressing, and mail delivery (see [1] for references).

Message Structure

A standard which describes the content and form of CBMS messages must have a model of a CBMS message from which to work. There are two models which may be used to describe the structure of messages.

- The "Header and Text" model divides a message into two portions, "header" and "text". The header is information about the message; the text is the information that the message is intended to convey. The header is further divided into a series of header fields, each with a particular usage and meaning.

- The "Fields" model states that a message is simply a series of fields. Each field has a particular usage and meaning. This model differs from the Header and Text model in that it doesn't attempt to differentiate between information about the message and the information the message is meant to convey.

The draft NBS message format standard uses the Fields model of message structure. It was chosen because as message systems have become more sophisticated, the difference between "header" and "text" has begun to become blurred. A message which represents a database entry is a good example of this. In this case, one would

be hard put to decide whether or not the date associated with the message was part of the header or the text. It describes the age of the information in the message, and might therefore be classified as header information. On the other hand, it might also be part of the text, since last update or access information may be included in an actual database entry. This type of message seems to violate the Header and Text model. The increasing use of such messages makes the Fields model the more viable model for a CBMS message.

MESSAGE FIELDS DEFINED BY THE DRAFT NBS MESSAGE FORMAT STANDARD

This section deals with the semantics portion of the draft NBS message format standard. It describes how the fields defined in the draft standard were chosen, the requirements for compliance with the semantics portion of the draft standard, and a number of the fields defined in the draft standard. A discussion of major issues in the design of the semantics portion of the draft standard closes the section.

Selection of Message Fields for Inclusion in the Draft Standard

The decision to include a field definition in the draft standard was based upon two criteria:

1. Common acceptance by current CBMSs
2. Ability to help support some CBMS function

Fields such as To, From, and Text are created and recognized by nearly every CBMS on the market. Occasionally implementations differ on field naming conventions. Since the draft standard does not specify how fields are to be labeled for presentation to users, differences such as the use of "Re:" instead of "Subject:" are ignored by the draft standard, and only one field which conveys the subject of the message is defined.

Message fields usually exist to support particular CBMS functions. They do so by supplying information which is necessary to perform these functions. The draft NBS Message Format Standard was designed to reflect the range of CBMS functions supported by current message systems used by business and government. The intent was to identify as many generally useful, well-defined functions as was reasonable, and to provide common mechanisms (fields) to support them as necessary. Each function may be supported by the use of one or more fields. Some examples of CBMS functions and the information they require are:

- Message creation and posting - Identify who is responsible for the message and for whom it is intended.
- Message reissuing - Identify a set of new recipients for an "old" or received message and who passed the message on to them. This function may be implemented in either of two ways, both supported by the draft standard.
- Reply generation - Identify who should get replies to a message.

- Cross-referencing - Identify "this" message. Provide citations to other messages which this one may obsolete, reference, or be a reply to.

- Life span functions - Record events such as posting, indicate when a message is "in effect", set time for "tickler" or alarm clock functions.

- Requests for recipient processing - Identify who should receive any return receipts, identify a list of recipients for serial distribution of a message.

All of the functions listed above may be supported using fields defined by the draft standard. There are a number of less common fields defined in the draft standard because they can be used to support a CBMS function which was judged to be of potential use in most systems, although not necessarily widely implemented. An example of such a field is Return-Receipt-To. It supports the RETURN RECEIPT function by requesting (by its existence) that the individuals which it identifies be notified of the receipt of the message.

Many of the field definitions were drawn from RFC 733.[2] Others, not as well documented, were drawn from experience instead of the literature.

Compliance

There are thirty-nine fields defined in the draft NBS message format standard. No one CBMS now supports, or is expected to support, all of them. In order to avoid placing an unnecessary burden upon CBMS implementors and vendors, the fields were divided into three compliance categories:

REQUIRED — These fields must be present in all messages and must be processed by message receiving programs as defined by the draft standard. Fields placed in this compliance category provide a means for checking the authenticity or integrity of a message.

BASIC — These fields need not be present in all messages but must be processed by message receiving programs as defined by the draft standard. Fields placed in this compliance category provide information vital to the correct interpretation of a message.

OPTIONAL — These fields need not be present in all messages and may be ignored by message receiving programs. The exact meaning of "ignored" is not specified by the draft standard. In general, a CBMS must recognize the existence of an optional field (that is, optional fields should not cause errors) and must not process the field in a manner contrary to the semantics defined for that field by the draft standard. All fields which were neither basic nor required were placed in this compliance category, including a few (such as Subject and Cc) which are used by almost every CBMS on the market.

Summary of Fields Defined by the Draft Standard

This section provides an overview of the message fields defined by the draft NBS Message Format Standard. The fields are grouped by compliance category, as described above.

Required message fields
There are only two required fields, From and Posted-Date. They identify who is responsible for the message and when it was posted. These fields were made mandatory in each message in order to give a reader a starting point to work from if there is any doubt about the message's authenticity or meaning.

Basic message fields
There are only three basic fields, Reply-To, Text, and To. Reply-To identifies the individuals to whom replies should be sent. Text contains the primary message content, that is, the primary information or document which the message is intended to convey. To identifies the primary recipients of the message.

Reply-To was included in this compliance group to ensure that if a user of one complying CBMS specifies where replies to a message should be sent, then his request will be treated uniformly by users of other complying CBMSs. Without such assurance the Reply-To field would not be useful. The individual identified in the Reply-To field of a message might not receive all of the replies; some still might be sent to whoever was identified in the From field. (These replies would come from users of CBMSs which did not recognize Reply-To.) The individuals identified in From and Reply-To field would always have to coordinate to make sure that the "Reply-To" person got all the replies.

Text and To were included in this compliance group because of their functions and universal use. If one of these fields were discarded from a message, the intentions of the message's originator might be lost or misconstrued. Every CBMS known to the authors of the draft standard uses some form of these fields.

Optional fields
The following paragraphs describe a few optional fields worthy of special note.

Assigned-By, Assigned-Date, Assigned-For-Action, and Assigned-For-Info all may be used to support message assignment. Redistributed-By, Redistributed-Date, Redistributed-To, and Redistributed-Cc all may be used to support message redistribution. Both assignment and redistribution are special cases of message reissuing (sending one or more "received" messages to a new set of recipients). Assignment is used to make someone else responsible for a message. Redistribution is used to make others aware of the complete and unaltered contents of a message. Redistribution does not imply a transfer of responsibility.

The Author field has been provided to differentiate between the individual who wrote the message and the individual who has taken responsibility for it.

Circulate-To and Circulate-Next may be used to support message circulation, which is the serial distribution of a message to its

Summary of Fields Defined by the Draft Standard

This section provides an overview of the message fields defined by the draft NBS Message Format Standard. The fields are grouped by compliance category, as described above.

Required message fields
There are only two required fields, From and Posted-Date. They identify who is responsible for the message and when it was posted. These fields were made mandatory in each message in order to give a reader a starting point to work from if there is any doubt about the message's authenticity or meaning.

Basic message fields
There are only three basic fields, Reply-To, Text, and To. Reply-To identifies the individuals to whom replies should be sent. Text contains the primary message content, that is, the primary information or document which the message is intended to convey. To identifies the primary recipients of the message.

Reply-To was included in this compliance group to ensure that if a user of one complying CBMS specifies where replies to a message should be sent, then his request will be treated uniformly by users of other complying CBMSs. Without such assurance the Reply-To field would not be useful. The individual identified in the Reply-To field of a message might not receive all of the replies; some still might be sent to whoever was identified in the From field. (These replies would come from users of CBMSs which did not recognize Reply-To.) The individuals identified in From and Reply-To field would always have to coordinate to make sure that the "Reply-To" person got all the replies.

Text and To were included in this compliance group because of their functions and universal use. If one of these fields were discarded from a message, the intentions of the message's originator might be lost or misconstrued. Every CBMS known to the authors of the draft standard uses some form of these fields.

Optional fields
The following paragraphs describe a few optional fields worthy of special note.

Assigned-By, Assigned-Date, Assigned-For-Action, and Assigned-For-Info all may be used to support message assignment. Redistributed-By, Redistributed-Date, Redistributed-To, and Redistributed-Cc all may be used to support message redistribution. Both assignment and redistribution are special cases of message reissuing (sending one or more "received" messages to a new set of recipients). Assignment is used to make someone else responsible for a message. Redistribution is used to make others aware of the complete and unaltered contents of a message. Redistribution does not imply a transfer of responsibility.

The Author field has been provided to differentiate between the individual who wrote the message and the individual who has taken responsibility for it.

Circulate-To and Circulate-Next may be used to support message circulation, which is the serial distribution of a message to its

recipients. It is done by including a distribution list as part of the message; when the message is first sent, it is sent to the first recipient on the distribution list. The recipient, or someone the recipient delegates, sends the message on to the second recipient on the list, perhaps after commenting on or adding to the message. This continues until all recipients on the distribution list have received the message.

A message originator wishes to circulate a message to recipients A, B and C. The originator includes the following fields in the message:

```
To:              A
Circulate-To:    A, B, C
Circulate-Next:  B, C
```

When recipient A or somebody A delegates causes the message to be further circulated, the message is sent to the first address in the Circulate-Next field, and that name is removed from that field:

```
To:              B
Circulate-To:    A, B, C
Circulate-Next:  C
```

B now sends the message on to its final recipient:

```
To:              C
Circulate-To:    A, B, C
```

Figure 1
Example of Message Circulation

The Circulate-To field contains the distribution list, which indicates to whom the message should be circulated, and the Circulate-Next field indicates all those in the Circulate-To field to whom the message has not yet been circulated. See Figure 1 for an example of message circulation using these two fields.

Reissue-Type is required to be found in messages which have been reissued using the syntactic method of message reissuing (see "Design Issues", below. It indicates why the message was reissued (assignment, redistribution, other type of reissuing).

Start-Date, End-Date, and Warning-Date may be used to indicate when a message takes effect, when a message loses effect, and when a message recipient should be reminded about the message or its contents.

Design Issues

The one major issue in the design of the semantics section of the draft NBS message format standard was the treatment of message reissuing.

Message reissuing can be handled in one of two ways, syntactic and semantic. The syntactic method, also called encapsulating, embeds the entire original message inside a new message.[3] The semantic method modifies the existing message by adding new fields to the message prior to sending the message out again.[4] If the message is being redistributed then Redistributed-By, Redistributed-Date, and Redistributed-To (or Redistributed-Cc) fields are added. If the message is being assigned the Assigned-By, Assigned-Date, and Assigned-For-Action (or Assigned-For-Info) fields are added.

Encapsulating adds structure around the original message, making it into a Message data element in a new message. This allows any part of it to be easily extracted[5]. The message which contains a message which is being reissued uses the From and Posted-Date fields to indicate who is responsible for the reissuing and when it took place. It uses the Reissue-Type field to indicate what kind of reissuing (Assignment, etc.) was being used. Figure 2 shows the contrast between the two methods of reissuing.

Syntactic Reissuing (Encapsulating)	=	Semantic Reissuing (Adding fields)

Contents of the Reissue-Type Field	Fields Used	Semantically Equivalent Fields
"Assigned"	To	Assigned-For-Action
	Cc	Assigned-For-Info
	Posted-Date	Assigned-Date
	From	Assigned-By
"Redistributed"	To	Redistributed-To
	Cc	Redistributed-Cc
	Posted-Date	Redistributed-Date
	From	Redistributed-By

Figure 2
Fields used in Syntactic and Semantic Forms of Message Reissuing

The two methods differ in how they represent issuing information for several reissues of the same message. Encapsulating contains information about the sequence of reissuance automatically (because of the way a encapsulated message is constructed). Ordering information is not automatically available in the semantic method with its added fields, except through some ad hoc fashion such as ordering of fields. The syntactic method requires slightly more processing power to extract the embedded message for whatever

processing is desired.

To illustrate how ordering information may be lost when using semantic reissuing, consider the following set of fields for a message that was redistributed twice:

```
...
Redistributed-To: Smith, Jones
Redistributed-By: Green
Redistributed-Date: 5 July 1980 16:32-EST
Redistributed-To: Johnson
Redistributed-By: Smith
Redistributed-Date: 6 July 1980 10:29-PST
...
```

The assumption, on cursory examination of the above, is that Green redistributed the message to Smith, who then redistributed it to Johnson.

Because the draft NBS standard does not impose field ordering or presentation conventions, the set of fields in the above example might, instead, have been ordered as follows for display to a user.

```
...
Redistributed-To: Johnson
Redistributed-To: Smith, Jones
Redistributed-By: Green
Redistributed-By: Smith
Redistributed-Date: 5 July 1980 16:32-EST
Redistributed-Date: 6 July 1980 10:29-PST
...
```

In this case, there is no way to determine who sent what to whom and when.

Despite this shortcoming, the semantic method of message reissuing was included in the draft standard because it is the method used by all or nearly all of today's CBMSs. This is in keeping with the decision to maintain compatibility between the draft standard and existing implementations. The syntactic method was included in order to provide a more powerful alternative which did not suffer from the shortcomings of the semantic method.

A REPRESENTATION SCHEME FOR MESSAGES AND THEIR COMPONENTS

The message syntax, or representation scheme, defined by the draft NBS message format standard was designed to support the representation of the messages read and generated by today's CBMSs, to be economical to use, and sufficiently flexible to allow graceful extension, especially in the direction of including multi-media objects in messages. In addition to supporting the representation of messages and their component parts, the syntax supplies mechanisms for indicating the use of data encryption or compression on any portion of a message.

The following sections discuss the representation scheme in terms of the major inputs into its design, an overview of the representation scheme itself, and the major issues in its design.

Design of the Syntax

The draft NBS Message Format Standard defines a series of data elements to be used for representing messages, and their components. The draft standard was designed to

- allow the representation of any object that may be found in conventional, text-based messages, and
- be sufficiently flexible to allow the representation of non-textual objects (such as facsimile images) which are beginning to be dealt with by more advanced CBMSs.

Objects found inside conventional messages include:

- Fields
- Dates and times
- Identification of message originators or recipients
- Unique identifiers (such as message-ids)
- Text strings

More complex messages may contain:

- Numbers
- Bit strings
- Boolean values
- Additional structuring information (below the level of a field)

The draft standard defines representations for all of the above objects. The representations were designed to be economical to use and sufficiently flexible to allow future extensions to the draft standard. An economical representation is the result of a favorable trade-off between the desire to both at once be concise and require little computation to interpret the data. A flexible representation provides general mechanisms which may continue to be used as the data to be represented grows in variety and complexity. Use of a general representation scheme allows CBMSs to exchange data with other office automation and general computation tools.

Economy is necessary because the costs associated with data storage and transmission join with speed and cost of computation in making a CBMS a viable commercial product. A flexible representation is necessary in order to avoid having the draft standard become quickly outdated by rapidly advancing CBMS technology.

These design criteria resulted in the decision to use a machine-readable (non-text) representation scheme. (For further discussion of that decision, see "Human versus Machine Readable Encoding Schemes", below.)

In addition to meeting the design criteria of economy and generality, the data elements were designed to be self-identifying. That is, data elements have consistently encoded in them sufficient information about their format and length to allow a CBMS which is unfamiliar with a particular data element to at least determine its extent.

Self-identification is extremely valuable in the case of CBMS message exchange. CBMSs must have a handle on how to deal with unfamiliar data elements when they are encountered. One example of such a situation is the use of user-defined data elements. Another is an un-updated CBMS encountering a data element which is defined in an extension to the original standard. A CBMS should at least be able to identify and discard an object which it does not understand. This allows it to be able to process the data which follows the unfamiliar object.

The syntax defined by the draft NBS message format standard owes much to Postel's RFC 753 [8] and ISO and ANSI Header Format Standards. RFC 753 defines a set of data elements which are of use in message processing and other applications. The ISO and ANSI work provided a model for a self-identifying, easily parsed representation scheme.

The representation scheme suggested in RFC 753 differs from the syntax defined by the draft NBS message format standard in a number of ways.

1. RFC 753 uses the Header and Text model of message structure, while the draft NBS standard uses the Fields model.

2. The data elements defined by RFC 753 are not self-identifying. There is no way for a CBMS to figure out how long an unfamiliar data element is so that the CBMS can skip over it. The data elements defined by the draft NBS message format standard are self-identifying.

3. RFC 753 uses text strings (in English) to identify header fields while the draft NBS standard uses binary numbers. There is some small overhead involved in the use of text as opposed to numeric values for identification purposes, but the real difficulty lies in the use of such a scheme in an international arena. It is desirable for a national standard to be identical to (or at least highly compatible with) internationally accepted standards with the same scope. It is highly unlikely that the use of English (or any other language) text strings to identify objects would be acceptable in an international message format standard.

Features of the Syntax

This section describes the message syntax defined in the draft NBS Message Format Standard. It first discusses the range of data elements defined in the draft standard. Then it describes the compliance requirements associated with the syntax. Finally the representation scheme itself is discussed.

Data elements

The set of data elements defined in RFC 753 served as a starting point in determining a set of data elements to be defined in the draft NBS Message Format Standard. The draft NBS standard defines a few additional data elements which are not found in RFC 753:

- Compressed - This data element contains a bit-string which is a data-compressed representation of any data element.
- Date - This data element contains a text string which is a representation of a date as defined by the pertinent ANSI standard [14].
- Encrypted - This data element is similar to Compressed. It contains a bit-string which is an encrypted representation of any data element.
- Field - This data element is used to represent a message field. Message fields have semantics defined in the draft standard.
- Message - This data element contains a series of fields which comprise a unit of communication.
- Property - This is an element in a property-list.
- Set - This data element is similar to a sequence, or list. It differs in that there is no ordering implied upon the elements in a set.
- Unique-ID - This data element contains a bit-string, text-string, or integer which may be used as a unique identifier within the domain of interconnected CBMSs.

The choice to include the above data elements in the draft standard was the result of a decision to trade "layering" against compatibility with existing CBMS implementations. This choice is further discussed in a later section, "The Role of Layering".

The draft NBS standard also defines a User-Defined data element which may be used by users and vendors to represent private or special-purpose data elements. An Extension data element has been set aside for future use.

Compliance

Many of the data elements defined in the draft NBS standard are not used in simple CBMS implementations. In order to avoid burdening implementors and vendors of simpler CBMSs, data elements were divided into two groups, basic and optional.

BASIC	These elements must be processed as defined in the draft standard by all message receiving systems.
OPTIONAL	These elements need not be processed by message receiving systems.

There are five basic data elements, ASCII-String, Date, End-Of-Constructor (see the next section for a description of this data element), Field, and Message. All other data elements are optional.

Overview of the Representation Scheme
The draft NBS Message Format Standard classifies data element types as either primitives or constructors. Primitive data elements, such as ASCII-String, are basic building blocks. One primitive data element, End-Of-Constructor, is used only as a structural element within constructor data elements and has no meaning by itself. End-of-Constructor is used to provide an end marker for constructor data elements that do not have an explicit length.[6] Any other use is not valid syntactically.

Constructor data elements, such as Message or Sequence, contain one or more primitive or constructor data elements. Some constructors, such as Sequence, may be composed of any other data element. Some, such as Message, may contain only certain data elements (in this case, only Fields or other Messages).

Two formats, fixed length and variable length, are used to encode the data elements. Some primitives are fixed length and some are variable length. All constructors are encoded as variable length.

Any data element may have associated with it a Property-List, which contains properties such as a Printing-Name or one or more Comments. A mechanism to support vendor- and user-defined properties has been supplied by the draft standard, as well as a mechanism to extend the list of property identifiers.

Printing-Names are used to provide labels that can be displayed along with their respective data elements. For example, a message originator may use a Printing-Name property to request that the To field of a message be labeled "Distribution:" when it is printed by its recipients. (This assumes that the CBMS receiving the message is capable of this processing.)

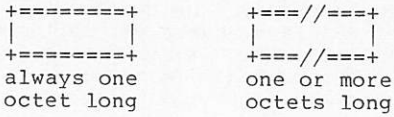
The Comment property is used to allow comments to be associated with any data element without affecting its actual contents. For example, someone reviewing the text of a message could add the comment "This looks good" to the Text field without either altering the body itself or adding a separate comment field.

Encoding Scheme
All data elements consist of a series of components.

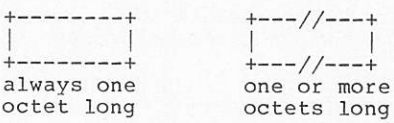
- Identifier octet (identifying particular type of data element)
- Length Code (specifying number of octets that appear following it in a data element)
- Qualifier (supplying additional identifying information)
- Property-List component (a Property-List data element containing Property data elements)
- Data Element Contents (containing the actual data of the data element)

These components always appear in this order. Not all components are present in all data elements but the components that are present maintain this relative order.

The syntax may be summarized by two diagrams. In these diagrams, components of a data element which are required to be used appear as follows. (The double border signifies "required".)



Optional components of data elements are represented as follows. (The single border signifies "not required".)



The first octet in a data element is the identifier octet. In these diagrams, all eight bits of the identifier octet are shown. Bits with fixed values show the fixed values as 1s and 0s. Bits with variable values are shown as x's and y's.

The first bit in an identifier octet is the P-bit. Its value indicates whether a data element contains a property list. (A P-bit value of 1 indicates the presence of a property list.)

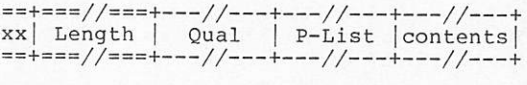
The second bit in an identifier octet indicates whether the data element is fixed or variable length. (Fixed length is indicated by 0; variable length by 1.) The remaining six bits contain the rest of the identifier.

In diagrams of syntax the data element components are labeled according to their class.

Component Class	Label
Length code	Length
Qualifier	Qual
Property-List	P-List
Contents	Contents

Fixed-length data elements must follow this form. The data element identifier contains the length of the Data Element Contents. The length appears in the bits represented by "yyy".

Variable-length data elements must follow this form.



The value of the Length component is the total number of octets

following the length code octet in the data element.

Design Issues

There were two major issues encountered in the design of the syntax of the draft NBS Message Format Standard, human vs. machine readable encoding schemes, and the extent to which "layering" was used in defining the representation of objects found in messages. These design issues are elaborated upon in the following sections.

Human versus Machine Readable Encoding Schemes
Human-readable (text stream) versus machine-readable encoding schemes have been a subject of debate in message processing circles for some time. A major decision in the design of a syntax for the draft NBS Message Format Standard was the choice between these two approaches.

The arguments in favor of text messages are:

- Robustness - If one or more characters are lost or garbled it is usually still possible to interpret the rest of the message.
- Human readability - Messages can be read by people without any translation. This allows the implementation of extremely simple (and possibly null) message display software.
- Existing usage - Many of today's CBMSs use a representation scheme which is mostly or completely text-based.

The arguments favoring a machine-readable format include:

- Simplified parsing - Algorithms for parsing machine-readable data can be much simpler than algorithms for parsing the same data in human-readable form. The use of simpler algorithms usually leads to smaller, faster, and more reliable implementations for software of the same functionality.
- More concise representation - A machine-readable format can be more concise than a text-based format, especially when non-textual, large or complex objects are to be represented. Since storage and transmission contribute to the costs associated with CBMSs, concise representation is important.
- Greater flexibility - A machine-readable format allows the convenient representation of structured data. It also allows non-text data to be represented in its "native" form. This is important because it is often difficult or expensive to encode structured or inherently non-textual data in a text-based format. The amount of computation or storage involved may be sufficiently prohibitive as to rule out the inclusion of such data in text-based messages.

The draft NBS message format standard uses a machine-readable format. While a text-based representation scheme may be adequate

for simple computer-based message systems, it was felt that such a format would not meet the needs of the swiftly developing CBMS technology.

The Role of Layering

A "layered" representation scheme is one in which a small set of very primitive objects ("primitives") is defined. All other objects are built solely using primitives. These are known as "constructed" objects. A layered representation scheme is expected to be more general and flexible than a scheme in which all objects are defined as primitive objects.

The draft NBS Message Format Standard uses a representation scheme which is only partially layered. It defines a number of data elements (Message, Field, Date, Unique-ID, Property, Set, Compressed, and Encrypted) which might have been treated as constructed objects in a fully layered representation scheme. This was the result of a trade-off between layering and compatibility with existing CBMS implementations.

A fully layered scheme might use constructed versions of the above elements, using property-lists to unambiguously identify each constructed object independent of the context in which it is found. For example, a Field might be represented as a Sequence containing a Property-List, an integer, and a Sequence containing the data elements which form the Field's contents. The Property-List would contain a "Field" property. This would differentiate this Sequence from another Sequence which may contain both an integer and another Sequence but is not a Field.

A scheme such as this requires the use of Property-Lists (or some other identification mechanism) and a moderate amount of computation (digesting the Property-List) in order to identify the constructed object. These constructed objects also are somewhat larger (require greater storage) than their less-layered counterparts.

The requirements for the use of Property-Lists and additional computation to identify constructed objects tipped the scales against the use of a fully layered representation scheme by the draft NBS Message Format Standard. No existing commercially available CBMS implements property-lists. The additional computation would probably be an impractical addition to many of the smaller, simpler CBMSs. The need to maintain compatibility between the draft standard and today's commercially available CBMSs resulted in the compromise approach to layering used in the draft standard.

CONCLUSIONS

The draft NBS Message Format Standard defines a set of set of standard message fields which may be used to convey specific information often found in CBMS messages. It also provides a machine-readable format for the representation of CBMS messages as they leave or enter a CBMS. However, use of the draft standard alone will not insure that two dissimilar CBMSs can be successfully interconnected. Additional development of the message format is necessary. A set of message transfer protocols and a standard CBMS envelope are needed to allow messages to be reliably transferred between systems. A message presentation standard is required to

allow messages to be presented to their recipients in the form their originators intended. The following paragraphs discuss how these other CBMS standards and protocols relate to a CBMS message format standard, and suggest some topics for future CBMS message format standardization efforts.

The exact nature of message posting, relay, and delivery protocols is only loosely coupled to the format of the messages being transferred. The protocols must only be able to transfer the messages transparently. On the other hand, the nature of a CBMS envelope has much in common with the message it contains. The types of information which can be found "on" a CBMS envelope are the same or very similar to the types of information found in a CBMS message. Both contain originator or recipient identifiers, dates, unique identifiers, and so forth. Envelopes and messages are structured to allow the programs which examine them to easily extract the specific information needed to perform a given function. For example, a Mail Service Agent reads the envelope in order to determine to whom a message should be delivered and a User Agent reads a message in order to determine where replies should be directed. For these reasons, standards for CBMS envelopes and messages might be very similar. The large potential savings in software development and maintenance costs practically dictates that the two standards be designed to be as compatible as possible.

A message presentation standard provides a set of rules for a User Agent to follow when a message is examined by its recipient. A simple message presentation standard might deal with things like the default order in which message fields should be examined and presentation rules for text contained in messages. A message presentation standard might allow an originator very fine control over how all or part of a message is presented by defining a set of presentation controls which may be embedded as data in the message. A message format standard should not constrain the design of a message presentation standard by precluding the use of such mechanisms. Instead, the message format standard should contain design "hooks" which can be used by the developers of a message presentation standard.

As already noted, the draft message format standard does not completely specify the representation of all parts of a message. For example, it does not state how message originators and recipients should be identified and how this information should be represented. It contains no specification for the representation of unique identifiers or multi-media objects. In addition, the draft message format standard does not provide a mechanism for data sharing. (Data sharing is the multiple use of a data element via references to a single copy.)

A standard for the representation of message originator and recipient names and addresses is absolutely necessary for the successful interconnection of different CBMSs. Standardized unique identifiers will become more important as CBMSs grow in function and complexity. Likewise, a standard for the representation of multi-media objects is not immediately required, but will be needed as CBMSs which handle messages containing digitized voice, facsimile, and other media become more available. A standard mechanism for data sharing also is not immediately necessary, but will become more desirable in time. Data sharing will offer storage and transmission

economies for large multi-media objects. It also may be used by CBMS envelopes, which will contain duplicate copies of some information found in messages.

As these standards are developed there are two principles which should be kept in mind. The designer of a CBMS message format standard should consider the relationship between the message format and other CBMS standards. The standard should be amenable future extension as CBMS technology advances. A thoughtful approach to CBMS message format standardization will help to encourage the continued healthy growth of CBMS technology and the CBMS industry.

NOTES

[1]The major contributors to the technical design were D. Deutsch, R. Resnick, and J. Vittal. C. Mooers also assisted in the definition of message fields.

[2]One field defined in RFC 733, Sender, was slightly redefined in the draft NBS standard in order to simplify its use. The RFC 733 definition required the use of the Sender field when it would not be redundant with the From field; this requirement was removed.

[3]"Syntactic" refers to the use of syntactic (structural) elements to accomplish reissuing.

[4]"Semantic" refers to the use of semantic elements (fields) to accomplish reissuing.

[5]Note that a message can contain another message, and that message can contain another message, and so on to any depth of encapsulating. This can occur by reissuing a message several times using the encapsulation mechanism.

[6]All data elements are marked with their length. There is a special "indefinite" length code provided for use when it is not possible to determine the length of a constructor data element at the time it is being formed.

REFERENCES

[1] E. Feinler, J. Pickens, and A. Sjoberg.
Computer Message Services Bibliography.
Technical Report NIC-BIBLIO-791201, SRI International, Menlo Park, CA 94025, December, 1979.

[2] D. Dodds, D. Deutsch.
Message Systems on the ARPAnet: An Evolution.
In Proceedings, IEEE Communications Techniques Seminar. IEEE, March, 1979.

[3] A. McKenzie.
File Transfer Protocol.
RFC 454, Network Information Center, SRI International, Menlo Park, California, February, 1973.

[4] Task Group X3S33 on Data Communications Formats, ANSI Subcommittee X3S3 on Data Communications.
Third Draft Proposed American National Standard for Heading Format Structure for Code Independent Communication Headings.
ANSI document X3S37/80-01, Computer and Business Equipment Manufacturers Association, 1980.

[5] ISO/TC97/SC6 Data Communications.
Second Draft Proposed Communication Heading Format Standard.
ISP/TC97/SC6 N 1948, ISO International Organization for Standardization Organiztion Internationale de Normalisation, September, 1979.
Secretariat: USA (ANSI).

[6] D. Deutsch, R. Resnick, J. Vittal.
Specification of a Draft Message Format Standard.
BBN Report 4486, Bolt Beranek and Newman, Cambridge, Massachusetts 02238, September, 1980.
Also published under the same title by the U.S. National Bureau of Standards as Report ICST/CBOS - 80-2.

[7] David H. Crocker, John J. Vittal, Kenneth T. Pogran, D. Austin Henderson, Jr.
Standard for the Format of ARPA Network Text Messages.
RFC 733, The Rand Corporation, Bolt Beranek and Newman Inc, Massachussets Institute of Technology, Bolt Beranek and Newman Inc., November, 1977.

[8] Jonathan B. Postel.
INTERNET MESSAGE PROTOCOL.
RFC 753, Information Sciences Institute, March, 1979.

[9] Jonathan B. Postel.
Out-of-net host addresses for mail.
RFC 754, University of Southern California, Information Sciences Institute, April, 1979.

[10] Debra P. Deutsch.
A Suggested Solution to the Naming, Addressing and Delivery Problem for ARPAnet Message Systems.
RFC 757, Bolt Beranek and Newman, September, 1979.

[11] Peter Schicker.
Naming and Addressing in a Computer Based Mail Environment.
Technical Report, Bell-Northern Research Ltd., Ottawa, Ontario, Canada, November, 1979.

[12] Peter Schicker.
The Computer Based Mail Environment: An Overview.
Technical Report, Bell-Northern Research Ltd., Ottawa, Ontario, Canada, December, 1979.

[13] John F. Shoch.
Inter-Network Naming, Addressing, and Routing.
In Proceedings, COMPCON. IEEE Computer Society, Fall, 1978.

[14] American National Standards Institute.
American National Standard Representation for Calendar Date and Ordinal Date for Information Interchange.
ANSI X3.30-1971, Computer and Business Equipment Manufacturers Association, 1971.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

A METHODOLOGY FOR THE SPECIFICATION OF A MESSAGE TRANSPORT SYSTEM

Gregor v. Bochmann

University of Montreal
Canada

John R. Pickens

SRI International
U.S.A.

A methodology for precisely specifying a distributed system application, message systems, is discussed. Message systems are typical of a class of applications in which more than the typical protocol or software specifications are needed. Such applications require specification not only of the protocol used by the different components of the system, but also of the overall systems model and of the services provided by the different parts of the system. The methodology contains multiple levels of abstraction, both horizontal and vertical, which describe the behavior and relationships of the different components of the system. The complete specification consists of an overall system model, the primitive operations for component interaction, declaration of data types, and rules for the temporal behavior of the components. The approach is illustrated by an example taken from an experimental multimedia message system and protocol currently being developed by DARPA.

1. INTRODUCTION

A growing variety of message systems are presently being designed and implemented. A problem which is reaching major proportions is the interworking between these different systems.[1] In order to make interworking possible, the services provided by the different systems must be comparable. Even if they are not equal, certain adaptations may be devised for transforming one service into another. In any case, it is important to have a precise specification of the service(s) provided by each message system.

As long as the message system consists of one protocol layer with a well defined transmission service for the exchange of protocol data units between the different message system modules, the specification of the service to be provided to the user of the message system may be left relatively vague, although the user may have difficulty adapting to this vagueness. The situation at present is similar to that of the case of the HDLC link layer protocol, which was originally developed without a precise specification of the link layer service. When the modules that used this service had to be defined precisely (for example the network layer over HDLC) then the provided service had to be known precisely.

We note that most existing message systems have essentially one layer of "message protocols" which are responsible for delivering messages to the correct destination. Considering more advanced message system architectures [5, 2] which may include modules for handling name-to-address binding, database servers,

[1] See [8, 7] for typical models of and functions within such systems

media transformation servers, etc., we have a situation where the basic message transport protocol is related to a number of additional system modules that are involved in these different services. It is therefore important that the service provided by the different modules be specified precisely, and at multiple levels of abstraction, so that the operation of other modules can rely on this service.

The object message system of this effort is that of the Draft Internet Message Protocol defined by DARPA [9, 10]. This protocol leaves certain aspects of the provided message service open. This leads to some difficulties, as argued in section 3 of this paper. We then give in section 4 an outline of a possible service specification consistent with the protocol, and in section 5 a formalized specification of this protocol. In section 6 we make a few comments on the Draft Internet Message Protocol on points that came to our attention during our work on its formal specification.

The emphasis of this paper is on the specification technique, and the importance of precise service specifications. We feel that the approaches reported in this paper may be applied to other message system designs as well as the design of other network based distributed systems.

2. APPROACH

Our specification approach proceeds in the spirit of the evolving specification approaches of the ISO Working Group on Formal Description Techniques [4]. Our specification, therefore, includes a description of the message system "model", followed by the service specification and then by the protocol specification. An interface specification can also be given, though we have not done so, but we recommend it for completeness.

This approach differs from the ISO approach primarily in adopting a hierarchical abstract machine view rather than a layered protocol view. Further study may show the differences to be more a matter of degree than substance.

For specifying the behavior of the modules within such an advanced message system, we have found the model of hierarchical protocol layering to be of limited utility. Rather, a model using abstract machines and step-wise refinement--leading to the specification of machines inside of given machines--has seemed more applicable and flexible. This model has been applied in this paper.

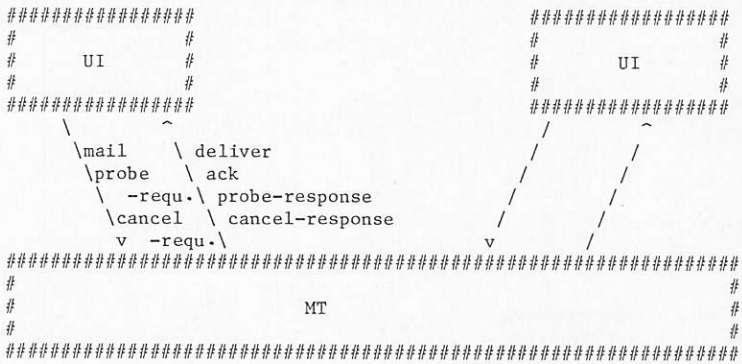
For the specification of a given abstract machine, we consider two parts: (a) the definition of the abstract interfaces over which the machine interacts with other machines. Each abstract interface is defined by the list of different interaction events that may occur at the interface, and possibly some (local) constraints on the order in which they may occur, and (b) constraints (global to the whole machine) determining the possible order and parameter values of the interactions that occur at the different interfaces of the machine.

For the definition of the interaction events and the domain of their parameter values, we found the data type definitional facilities of Pascal and similar languages very useful. For the examples in this paper we have adopted the notation of ADA [1]. For the specification of the global constraints (there are no local constraints in the example considered), we use in this paper some "formalized English". This part of the examples could be specified more formally using either a state-oriented specification language, or a formalism based on execution sequences, possibly including temporal logic [6, 11, 3] or other approaches [12].

3. MODEL OF THE MESSAGE SYSTEM

The message system consists of two basic abstract machines, the User Interface machine (UI), and the Message Transport machine (MT). The basic sequence of operations is as follows: A user interacts with his local UI to create a message. The user then instructs the UI to send the message. The UI, in turn, presents the message to the MT for routing and delivery. Eventually the MT presents the message to the destination UI and acknowledges this to the source UI.

The MT (which is internally distributed over different hosts, networks, and gateways) interacts with the UIs (which may reside in different hosts) as shown in the diagram below.



The specification of the message transport system is given in two steps of successive refinements:

- Global specification of the message transport machine (MT). This leads to the "service specification" of the "message transport layer". (We note that only part of this specification is implied by the document [9, 10].)

- Specification of a message transport module (MTM), which is an integral subcomponent of the message transport machine. This leads directly to the "protocol specification" of the "message transport layer".

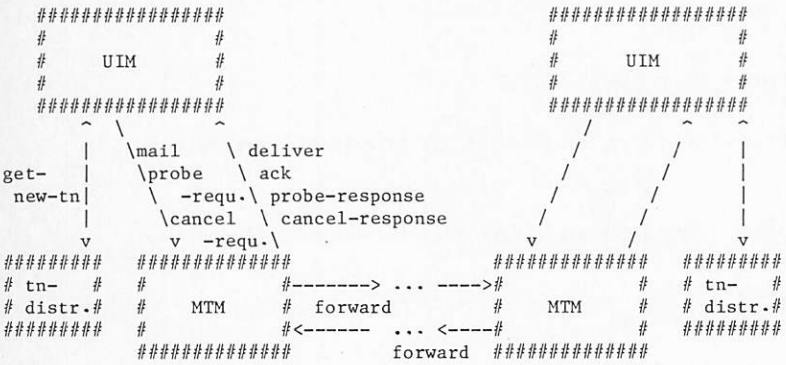
The User Interface machine may also be specified in successive refinements, revealing the UIM as the integral subcomponent. We choose to focus on the message transport machine, however. Note that our terminology differs from the reference [9] in that we use UIM/MTM rather than UIP/MPM. Note also that richer decompositions are possible for alternate advanced message systems such as [5, 2].

For each of these specifications we try to give only so much detail as is required for the logical compatibility of the UIMs and the MTMs. Therefore only the logical interaction primitives between the UIMs and the MTMs are specified. Conventions concerning possible grouping of several messages into "bags",

encryption and the format and coding of the exchanged messages are independent issues. They are discussed in a separate section. Similarly, the question of UIM-to-MTM and MTM-to-MTM communication (whether through shared memory in a local host or through some communication subsystem) is an independent issue. It is not specified here.

The specification of the MT is an overall description of the way in which the system interacts with UIs at the MT-UI interface. It corresponds to the service specification of [4], if the UI/MT relationship were modeled as a layered protocol hierarchy. This specification is distinct from the MTM specification given in section 5. The latter, which is a refinement of the MT specification, describes both the interaction between the MTM and the local UIMs and the interaction between MTMs for the forwarding of messages.

The MTM-specification does not explicitly include the "end-to-end" properties of the MT specification. It is assumed that these properties can be derived from the MTM specification. Such a derivation constitutes a design verification of the MTM specification (it may also be called a verification of the "message transport protocol").



2

One tn-distributor exists per host

We note that the document [9, 10] concentrates on the MTM specification. However, the following points indicate the importance of a precise and logical MT specification.

- Certain design questions can be answered independently of the message forwarding protocol: How does the UIM indicate the identity of a message to be canceled? When is an acknowledgement of a forwarded message generated? Does the UIM obtain error status and MTM trace information when it receives an acknowledgement?

---

2 The interaction of the tn-distributor with the UIM is different from what is specified in [9, 10]; see section 4.

- Interworking between different message systems seems to imply an agreement on some minimal message service to be provided by all systems. This minimal service may be defined without speaking about message forwarding. It is clear, however, that this service will have an impact on the message forwarding protocol.

- Within the context of a layered system structure including several protocol layers the service specification of a given layer is the basis for the operation of the layer above, which uses the service. It represents also the requirements that must be satisfied by the protocol layer below, which should provide this service.

- The MT specification is required for a verification of the message forwarding protocol.

4. MT SERVICE SPECIFICATION

The MT specification given below is strongly influenced by the protocol definition of [9, 10], but certain design decisions have been made here in order to clarify the UI-MT interface. We mention in particular:

- Most information fields of forwarded messages are also exchanged with the UIs.

- In particular, the "tid" field (message identifier) is exchanged between the MT and the UI. As indicated in the figure above, the UI obtains unique identifiers from the tn-distributor module of the host and assigns these identifiers to the messages submitted to the MT. This simplifies subsequent interaction between the MT and the UI concerning the same messages, such as delivering acknowledgements or indicating probe or cancel responses. In order to "understand" these responses the UI needs to know the identifiers associated with the original requests.

We have structured the specification of the MT into the following three parts:

- specification of data types (corresponding to the "objects" of [9, 10]) which characterize the information exchanged between the message transport system (MT) and its users (UIs),

- specification of the service primitives that may be invoked for the exchange of information between the MT and one of its users, and

- global constraints on the execution of service primitives with different users (end-to-end properties).

4.1. DATA TYPES

The following is a specification, in ADA, of the structure of messages that are passed between the UI and MT. This structure is also utilized internal to the MT.

```
TYPE message IS
    RECORD
        tid : transaction-id ;
        destination : mailbox ;
        trace : MT-list ;
        command : (deliver, acknowledge, probe, response,
                   cancel, canceled) ;

        CASE command OF
            WHEN deliver =>
                options : delivery-options ;
                document : RECORD
                               header  : header-type ;
                               body : LIST OF body-portion ;
                           END RECORD ;
            WHEN acknowledge =>
                acknowledged-tid : transaction-id ;
                trail : MT-list ;
                answer : boolean ;
                CASE answer OF
                    WHEN  true =>
                        delivered-address : mailbox ;
                        delivery-mode : (regular, gendel) ;
                    WHEN false =>
                        reason : why-not-delivered ;
                END CASE ;
                how-delivered : delivery-options ;
                status : error-list ;
            WHEN probe => NULL ;
            WHEN response =>
                request-tid : transaction-id ;
                trail : MT-list ;
                answer : boolean ;
                CASE answer OF
                    WHEN true =>
                        address : mailbox ;
                    WHEN false =>
                        reason : why-not-delivered ;
                END CASE ;
                status : error-list ;
            WHEN cancel =>
                to-be-canceled :transaction-id ;
            WHEN canceled =>
                canceled-tid : transaction-id ;
                trail : MT-list ;
                answer : boolean ;
                status : error-list ;
        END CASE ;
    END RECORD ;
```

```
TYPE MT-list IS  ... ;
TYPE delivery-options IS SET OF (regular, forward, gendel);
TYPE why-not-delivered IS (no-such-user, no-such-host,
          no-such-network, address-ambiguous, mailbox-does-not-exist,
          mailbox-full,  ... ) ;
TYPE error-list IS
    RECORD
        error-class : (success, partial-results-returned,
            resource-unavailable, user-error, MT-recoverable-error,
            MT-fatal-error, user-abort-requested) ;
        error-string : text ;
    END RECORD ;
TYPE text IS  ... ;
```

4.2. SERVICE PRIMITIVES EXECUTED AT THE UI-MT INTERFACE

```
        3
    post  (m : message (command => deliver) )
    deliver (m : message (command => deliver) )
    ack (a : message (command => acknowledge) )
    probe-request (p : message (command => probe) )
    probe-response (r : message (command => response) )
    cancel-request (c : message (command => cancel) )
    cancel-response (c : message (command => canceled) )
```

We assume that these primitives are executed jointly by the two modules involved in the interface. The primitives post, probe-request, and cancel-request are initiated by the UI, the other primitives are initiated by the MT. The value of the parameter is determined by the initiating module. For the former, the trace field has the value empty; for the latter, the destination field has as its value the mailbox of the UI involved in the execution of the primitive.

4.3. GLOBAL CONSTRAINTS

The following rules specify constraints on the order of execution and on the possible parameter values of the primitives executed at different places within the message system ("end-to-end" properties). The first rule states that every message sent will be eventually delivered, while the second states that each delivery will be acknowledged. Similar rules must also be given for the probe and cancel primitives. As they stand, these rules are incomplete, since the handling of exceptional cases is not specified.

1. After the execution of the primitive post(m) by some UI and the MT, where m.destination = some value x, there will eventually be an

---

3 This line reads: There is a "post" primitive which requires a parameter of type "message" (defined in section 4.1) for which the "command" field has the value "deliver".

execution of a primitive deliver(m') by the UI with the address x and the MT, where m'=m except for the trace field; the first element of m'.trace = the ihn (internet host number) of the host where the post(m) was executed. This is the only case in which a deliver primitive will be executed.

2. After the execution of deliver(m') primitive by some UI and the MT, where the first element of m'.trace is the ihn y, there will eventually be an execution of an ack(a) with the host ihn = y, where a.acknowledged-tid = m'.tid, a.trail = m'.trace, and a.answer = true. This execution will be related to the mailbox of the UI which had previously executed the post(m) primitive with m.tid = a.acknowledged-tid. This is the only case in which an ack primitive will be executed.

4.4. ADDRESSING

```
TYPE mailbox IS
    LIST OF RECORD
        property : (ia, net, host, user, city, ... , phone) ;
        CASE property OF
            WHEN ia      => internet-address : integer ;
            WHEN net     => network-name : text ;
            WHEN host    => host-name : text ;
             ...
        END CASE ; END RECORD ;
```

5. SPECIFICATION OF THE MTM-LEVEL PROTOCOL

We use in the following the same specification approach and subsections as in the case of the MT specification above. The interactions of an MTM module are indicated in the second diagram above.

5.1. MTM INTERACTION PRIMITIVES

The following primitives are executed by the MTM:

- Primitives executed in relation with UIs (see section 4.2)

- forward (m : message) This forward primitive is used by the MTMS to forward messages towards the destination MTM. The "message" type is defined in section 4.1.

5.2. CONSTRAINTS

The following specifies constraints on the execution order and on the possible parameter values of the interaction primitives executed by a given MTM:

1. For each incoming message m with command = deliver (coming in through a post primitive from a local UI, or through a forward primitive from a remote MTM) do the following:

```
IF is-local(m.destination)
THEN execute deliver(m') with the corresponding UI where
        m'=m, except that m'.trace = m.trace appended with the
              host ihn (internet host number) of the MTM ;
     put an acknowledgement message a into the handling queue
     (for the same MTM)  where
              a.acknowledged-tid = m.tid,
              a.trail = m.trace, and
              a.answer = true ;
ELSE find-route(m.destination, next-MTM) ;
     execute a forward(m') primitive with the next-MTM
        where m' = m , except that
              m'.trace = m.trace appended with the ihn of the MTM ;

For each incoming acknowledgement ....
...
...
```

Note: The above specification is incomplete, because it does not consider the cases where the message cannot be delivered.

5.3. MTM-TO-MTM AND UIM-TO-MTM COMMUNICATION

Several additional layers of protocols must be specified for the MTM-to-MTM and UIM-to-MTM communication. This includes a specification of the underlying transport service, as well as protocols for message bagging, data compression, and encryption. Also the format and coding of the message type structure (see section 4.1) must be specified.

For the MTM-to-MTM communication these protocols must be standardized for all systems involved, while the protocols to be used between a given UIM and its MTM may be agreed upon by the two parties (often this will be conventions within one single computer operating system). The chosen protocols, however, must meet the requirements of the service specification.

5.4. MESSAGE BAGGING

Instead of initiating an interaction with a peer MTM for each message to be forwarded, several messages may be combined by an MTM into a bag, and a whole bag of messages may be forwarded between two MTMS (at an appropriate time) in a single interaction.

The same bagging strategy may also be used for the exchange of messages between a UIM and an MTM, although this is a question of implementation of the UIM-MTM interface, which has no impact on the UIM-to-UIM compatibility (i.e. a pair of UIM's are compatible even if one does not support bagging).

The bagging strategy described in document [9, 10] may be characterized by the following rule: Whenever a given data structure, say "some-data", is possibly to be shared between several messages the field "some-data" is replaced by a data structure of the form

```
shared-data : RECORD
                 sharing : boolean ;
                 CASE sharing OF
```

```
            WHEN true =>
                tid-to-share-from : transaction-id;
            WHEN false =>
                some-data : ... ;
    END RECORD ;
```

The complete specification which includes bagging may be found in [13].

6. COMMENTS ON THE INTERNET MESSAGE PROTOCOL

The following issues have arisen from our efforts to specify and evaluate the message transport system. We feel that the more formal and concise specification has helped to facilitate the detection of problem areas.

- The naming - addressing scheme of the protocol is host and network oriented, which corresponds to the addressing scheme of an interprocess transport protocol. Alternate naming schemes, such as organizational hierarchy, can be expected to be processed by both the UI and the MT.

- There is in general no information on the source of a message available to the receiving UIM and its MTM, except that the message header may include some source identification which, however, has only an informal meaning. This has the following consequences:

    1. The final MTM receiving an acknowledgement has some problem delivering it to the right UIM (unless it keeps a list relating the transaction numbers of the sent messages to the identity of the sending UIM's).

    2. A UIM module of user x may masquerade as user y (as seen by the receiver who reads the message header), unless the posting MTM checks the information inside the message header (which is usually not looked at by an MTM).

- The meaning of an acknowledgement is not clear as far as its "end-to-end" significance is concerned. It is not clear who generates the acknowledgement, the receiving UIM, the message server module of the UIM (which may be on a local network), or the last internet MTM through which the message travels. We assume the latter. In this case the message may not yet have reached the destination UIM when the acknowledgement is generated.

- It seems that for certain applications more flexible sharing arrangements within bags of messages would be useful, and should be foreseen. An example would be a structured message body consisting of several components where not all of these components are to be shared.

- The UIM needs to know the "tid" identifiers associated with the requests it passes to the MT. This is necessary for interpreting acknowledgements, and probe and cancel responses returned by the MT. It is also necessary if the UIM submits bagged messages.

- The dialogue between MTMs is simple, and probably should be expanded.

Two examples of needed inter-MTM dialogues which go beyond the basic "forward" primitive include change-of-address and flow-control. Change-of-address allows a message to be rerouted before transmission of the entire body, an important efficiency consideration. Flow-control may be needed in order for an individual MTM to restrict the message traffic flowing in or out of a specific path.

7. CONCLUSIONS

In this paper we have discussed a methodology for precisely specifying a message system. We have argued that such a specification must include not only the protocol(s) used by the different components of the system, but also the specification of the overall service provided to the users of the system.

In the context of advanced message systems with host-independent naming , specialized media etc. , the specification of each of the different system components must be given. The "model" of the message system which we use is not necessarily structured into a number of protocol layers; but rather a more flexible structuring principle is used where a given system modules may be subdivided into a number of submodules. The message transport system, for example, is considered as an abstract machine, the specification of which is the specification of the message service provided to its users. Subsequent subdivisions leading to more and more detail are then considered. One such division leads to the consideration of the MPM, the specification of which defines the message transport protocol.

The message service specification is given in terms of service primitives, which are also referred to by the protocol specification. Other advanced functions could be specified separately, defining additional modules.

Finally, we have found that our efforts in making the service and protocol specifications more precise have led us to a better understanding of that system, and to the discovery (or observation) of problem areas in the original design.

REFERENCES

[1]
Ichbiah, J. D., et al.
Preliminary Ada Reference Manual and Rational for the Design of the Ada Programming Language.
Sigplan Notices 14(6), June 1979.

[2]
Lebling, P. and S. Galley.
Advanced Message System Design Note.
Technical Report AMS.01.01, Programming Technology Group, Laboratory for Computer Science, Massachusetts Institute of Technology, April 21, 1980.

[3]
Hailpern, B., S. Owicki.
Verifying Network Protocols Using Temporal Logic.
Technical Report Technical Report 192, Computer Systems Laboratory, Stanford University, June 1980.

[4]
Guidelines for Specifications of Services, Protocols and Interfaces for OSI.
ISO/TC97/SC16/WG 1.

[5]
Levin, R. and M. Schroeder.
Transport of Electronic Messages Through a Network.
Teleinformatics 79 :29-33, 1979.

[6]
Manna, Z., A. Pnueli.
The Modal Logic of Programs.
Technical Report STAN-CS-79-751, Computer Science Dept., Stanford University, September 1979.

[7]
The Computer Based Mail Environment, An Overview.
IFIP Working Group 6.5 working paper submitted by Peter Schicker, Bell Northern Research Ltd., Ottawa, Canada.

[8]
Pickens, J.R.
Functional Distribution of Computer Based Messaging Systems.
In Proceedings of Sixth Data Communications Symposium, . ACM/IEEE, November 1979.

[9]
Postel, J. B.
Internet Message Protocol.
Technical Report IEN 85, USC Information Sciences Institute, Marina Del Rey, CA, March 1979.

[10]
Postel, J. B.
An Internetwork Message Structure.
In Proceedings of Sixth Data Communications Symposium, . ACM/IEEE, November 1979.

[11]
Schwartz, R. L. and P. M. Melliar-Smith.
Temporal Logic Specification of Distributed Systems.
August 1980.
Computer Sciences Lab, SRI International, submitted for publication.

[12]
Bochmann, G. V. and C. A. Sunshine.
Formal Methods in Communication Protocol Design.
IEEE Transactions on Communications 28(4):624-631, April 1980.

[13]
Bochmann, G.V. and J.R. Pickens.
A Methodology for the Specification of a Message Transport System.
Technical Report SRI/TSC/80-1, Telecommunications Sciences Center, SRI International, July 1980.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

ON RELIABLE TRANSPORT CONNECTIONS FOR EXCHANGING MAIL ITEMS

Friedrich H. Vogt

Hahn-Meitner-Institut G.m.b.H.
Berlin, Germany F.R.

Reliable transport connections are required to transfer mail items between Mail Service Agents in a computer based mail system. In following the Reference Model of Open Systems Interconnection end-to-end reliability enhancements are positioned in the Transport Layer. Thus, this enhancements are based on the requirements of the mail system on the one hand and the reliability provided by the network on the other hand. In this paper it is assumed that network connections are Virtual Circuits of X.25 type. Since a Transport Connection requires significant resources, Transport Connections should only be maintained while processes (in this case Mail Service Agents) are communicating. A mechanism is required for "connect" when needed and for "disconnect" after insuring that all transport-service-data-units have been exchanged. On the other hand it should be possible to "release" and "resume" network connections while processes are temporarily not communicating or if the network is temporarily unable to maintain the network connection. To provide both, an appropriate Transport Connection Protocol is proposed.

1. INTRODUCTION

The mail items that are communicated through a computer based mail system consists of an envelope and a content /15/. The content carries any digital coded information and can represent unstructured and structured data, voice data, facsimile data,.... The content is transparent to the mail system.

Users of mail systems may be persons connected via terminals to a local or public computer network providing a mail system. User agents are responsible for preparing and inspect mail items on behalf of the users of the mail service system. They are also responsible for posting and delivering mail items on behalf of the mail service user (see Fig. 1). Similar structures are adopted by other authors /16/.

Mail Service Agents are indicated as processes communicating with each other in using a certain kind of message transport service (e.g. /17/). The kind of message transport service proposed in this paper includes the possibility of temporary "release" and "resume" network connections during the lifetime of a transport connection. The basic ideas are described in /1/,/2/,/3/ and in /13/ according to the Reference Model /7/. A formal specification and verification of the protocol is given in /4/,/5/ and /6/. This type of transport service

may also be considered as an enhancement to the ECMA class 1 as described in /14/. It is assumed that this feature is required by computer based mailing systems. Furthermore, it is assumed that no difference should appear to users accessing the mailing service via a local network or via a public network.

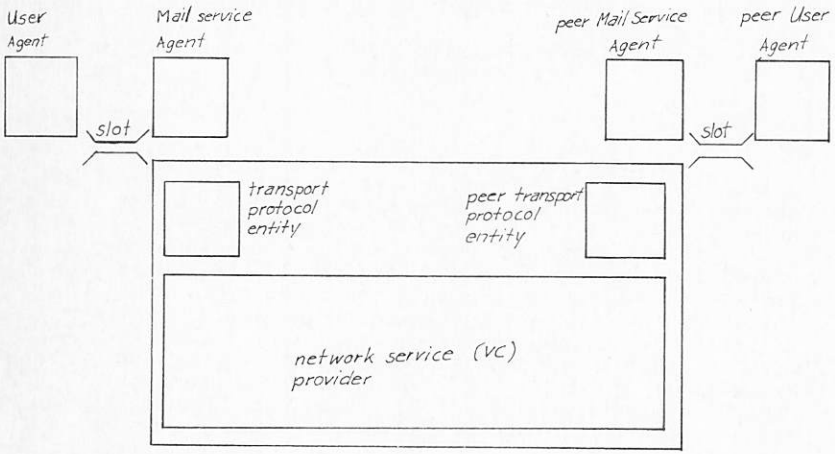


Figure 1

2. THE ENVIRONMENT CONSIDERED

The protocol defined by X.25 is associated with the first three layers of the Reference Model /7/. In HMINET II[1] the following configuration is considered (Fig. 2).

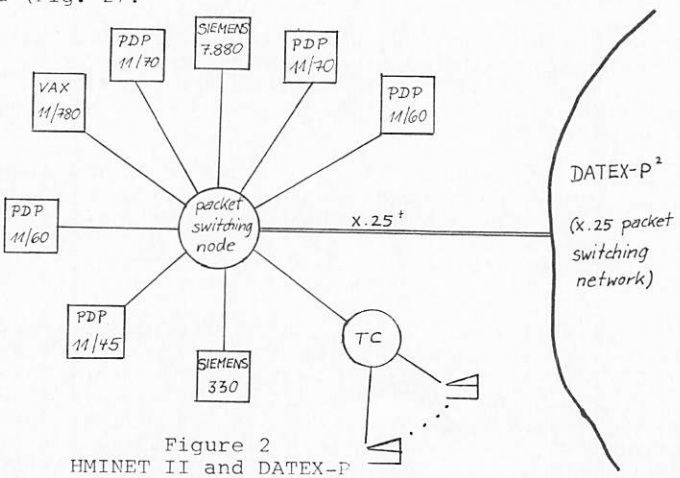


Figure 2
HMINET II and DATEX-P

It is assumed that the network addresses are structured as "X.25 DTE address + suffix". Mail Service Agents are identified by globally defined titles or by local titles in conjunction with the network address. A one-to-one mapping of transport connections onto network connections is assumed. Flow control at the interface from the network layer to their users (end systems) is essential. The flow control at the interface from the transport layer to their users (Mail S. Agents) is based upon this property (and not on an additional peer-to-peer flow control schema at the transport layer!).

3. NOTATION

In the following, a method for specifying a layer service based on abstract service primitives is used. A service primitive is an element of the provided service, making abstractions from the particular interface. A service primitive may be invoked (i.e., its execution may be initiated) by either side, the service providing and using layers.

Each service primitive is identified by a name which is written in capital letters. The execution of a service primitive may be initiated by the layer above (the user) or the entity providing the service. To each service primitive a related group of events is defined. Four types of events are considered, the request event (initiating a service to the lower layer), the indication event (to initiate a service to the service user), the response event (from the service user) and the confirm event (from the service provider). Whether parameter values may be provided (i.e. determined before the execution of the service primitive or in response to an indication event) by the request or response event, or obtained (i.e. determined during the execution of the service primitive) by the indication or confirm event.

For example,

INIT

represents the service primitive if name INIT. Since INIT.request and INIT.confirm establishes a network connection, INIT.indication and INIT.response corresponds to an incoming call in X.25.

4. ASSUMPTIONS ABOUT THE NETWORK LAYER SERVICE

4.1 General

The purpose of the network layer as stated in /7/ is to provide for the exchange of so-called network-service-data-units between end-systems containing application processes. It provides to the end-systems independence from routing and switching considerations with the establishment and operation of a given network connection (this includes the case where several transmissions resources are used in tandem).

4.2 Considerations about the interconnections of VC's

In the case of HMINET II where a local area packet-switching network (based on VC a la X.25) is interconnected with a public packet-switching network (based on VC a la X.25) interconnection has to be provided by the network layer. The intention is to provide a VC-service

(based on two VC's a la X.25) at end-system to end-system base. This requires a more general end-system addressing scheme.

Two possibilities are considered in order to provide end-system to end-system VC (based on interconnected X.25 networks). Fig. 3 represents an address structure simply derived from the hierarchical addressing scheme used in telephone networks.

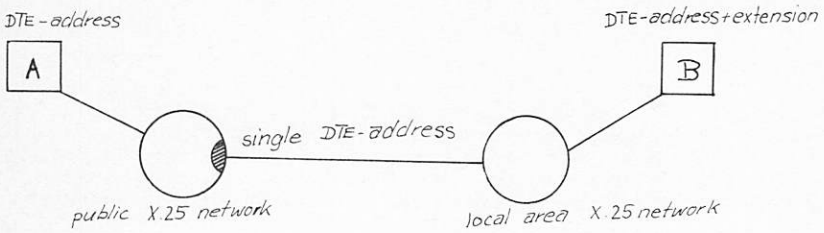


Figure 3

In this case the end-system addresses reflect the physical structure. The second possibility is represented, if all end-systems get an DTE-address according the numbering scheme of the public X.25 network and one or several interconnection lines are seen as multi DTE-address lines from the piblic X.25 network (Fig. 4).

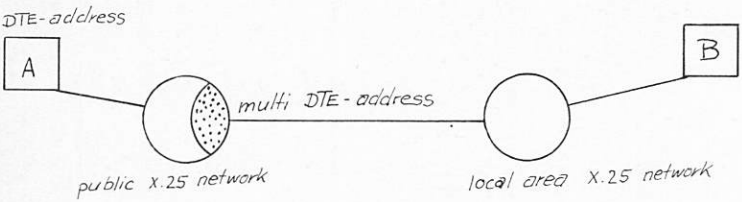


Figure 4

This would also include the interconnection of private X.25 networks via X.75, or the view that local area X.25 networks are as switching facilities logically an integral part of public X.25 networks and no standard interconnection protocol has to be considered.

The assumtion about the network layer service is that the VC service a la X.25 may be provided to the end-systems, even in the case that several interconnected X.25 networks are used in tandem.

4.3 VC Transmission Service Assumptions

The service described is used by the transport layer for the exchange of protocol-data-units between end-systems. It is assumed that a certain number of network connections are established between those end-

systems and dynamically allocated to the transport connections (connections between application processes) as needed. A network connection is either free, or it supports a single transport connection in the proposed transport connection protocol.

For this publication we assume that the basic transmission service is given by Virtual Circuits (VC) as provided through an X.25 interface /8/,/11/,/12/,/18/,/19/.

Since the transmission service is not exactly defined by the interface standard, we make the following relatively weak assumptions which (hopefully) are satisfied for all networks /8/,/18/,/19/, and also for inter-network Virtual Circuits:

- During a circuit reset, data and interrupt packets in transit may be discarded;
- Except for the above, messages in form of "user sequences" are sequentially transmitted without errors (Fig. 5);
- Interrupts are never duplicated;
- The interrupt confirmation does not necessarily have an end-to-end significance;
- An interrupt packet usually travels faster, but may travel slower than data packets;
- To each reset operation at one end-point of the circuit, there is a corresponding reset operation at the other end, but a subscriber-initiated reset operation at one end may correspond to a network-generated reset at the other end.

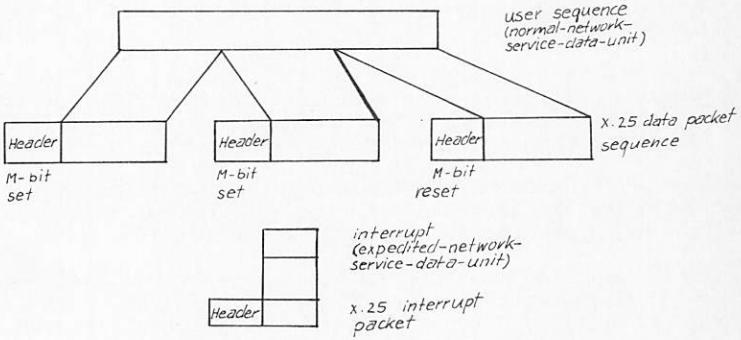


Figure 5

The data transmission service over VC is characterized by the following service primitives (initiation and clearing only for switched circuits):

INIT

Note, a network address used to identify uniquely a transport-entity

may consist of a X.25 DTE-address; a subaddress to X.25 DTE-address; and a subaddress-extension. The only specified address extension (in /9/) is for distinguishing between transport entities relating to X.29 (nullified) and others (DTE-DTE usage). The indicated version number is necessary if a transport-entity performs certain versions of transport protocols.

CLEAR

USER SEQUENCE

INTERRUPT

RESET

Note, for example, that the service primitive INTERRUPT represents the receipt of an interrupt (including the sending of the interrupt confirmation packet), and also the sending of an interrupt (including the receipt of the interrupt confirmation packet).

5. TRANSPORT-CONNECTION SERVICE AND MANAGEMENT

5.1 General

The establishment and termination of a transport connection is based on the Transmission Resource Allocation (TRA) service provided independently for each network connection (Fig. 6). TRA provides agreement between the two transport entities (using the network-connetion) on the allocation of the network connection to transport connections. It also supports the transport connection establishment and termination protocol by providing recovery of lost protocol data units (signalled by resets) and a timeout function during the phases of allocation and de-allocation of a network connection. The Transmission Resource Allocation function is architectured as a sub layer within the transport layer.

While the TRA service provides for the allocation of network connections between different transport entities, the CON service provides for transport connections between session entities which act on behalf of application processes (e.g. Message Service Agents).

Consequently, the CONNECT and DISCONNECT primitives of the CON service have many similarities with the services of the TRA service. However, there are the following differences:

- The CON service provides addressing up to the application process level: the destination for a connection to be established is characterized by a network-address (which identifies the responsible transport-entity) and a transport-address.

- The allocation of a network connection to an established transport connection may be explicitly controlled through the RELEASE and RESUME primitives.

- When the option of break recovery is used, the reliability of a transport connection may be much higher than that of an individual network connection (as long as alternative network connections are available between two transport entities).

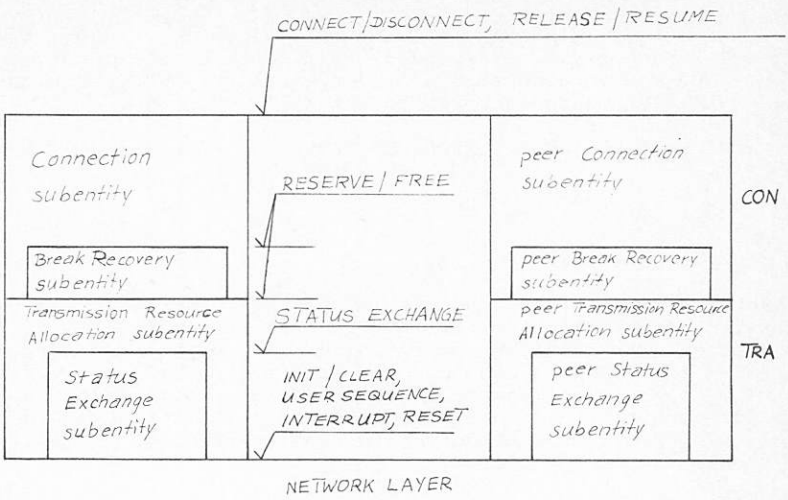


Figure 6

5.2 Transmission Resource Allocation Protocol

The analysis of the TRA protocol /6/ discovered that the previously proposed TRA protocol /3/,/4/,/5/, is not completely reliable in the case of network-service-data-unit losses and retransmissions.

A similar but reliable protocol is proposed by G.v.Bochmann in /6/. It is based on the so-called Status Exchange protocol providing the service of exchanging status-data. It opreates even in hostile environment with network-generated resets.

The Status Exchange protocol avoids the problems considered in /6/ by assiciating with each reservation interaction a sequence-number (similar conclusions have been made for connection establishment protocols working over a datagram transmission medium /10/). We assume that both entities know the current sequence number, and they use both the same algorithm (this is necessary to detect collision) to determine the next reservation number to be used in the next exchange. Therefore each entity can distinguish "old", "current", and "new" incoming protocol data units (by comparing the sequence number of the incoming p.d.u. with the current sequence counter of the Status Exchange entity).

It uses, for this purpose, the USER SEQUENCE and RESET service primitives provided by the underlying network layer for communication with peer transport entity at the opposite end of the network connection. It is assumed the network connection is previously established by the

entity using the Status Exchange service.

It is understood, although not explicitly described below, that during the "transmission phase" service-data-units and resets are transparently passed through to the sub-layer above in both directions. A CLEAR is always passed through to the sublayer above.

The service of the Status Exchange protocol is defined by the primitive

STATUS EXCHANGE

The protocol is defined in /6/.

The service of reserving and freeing network connections in conjunction with establishing or terminating transport connections (or in conjunction with releasing or resuming transport connections as well) is called the Transmission Resource Allocation Service. It handles the allocation of permanent and switched network connections between transport entities. The protocol is based on the Status Exchange protocol which provides a mechanism to recover from p.d.u. loss during the status exchange phase.

The following service primitives are defined:

RESERVE

FREE

5.3 Transport Connection Service and Protocol

The transport connection service (CON service) is resposible for establishing and terminating end-process to end-process transport connections. In order to use the flow control of the VC's, each VC is allocated to at most one transport connection.

The service of temporarily de-allocating the transmission channel, using the RELEASE primitive, is part of the CON service. A RELEASE primitive is also invoked when the transmission channel breaks (the network clears the VC). Note that after a RELEASE is performed, the connection is still existing, although in a "dormant" state. The RESUME primitive may be used to "reactivate" a released connection. or to recover from a broken transmission channel. It allocates a new channel to the resumed connection.

The connection oriented service primitives are defined by:

CONNECT

DISCONNECT

RELEASE

RESUME

Two problems have to be considered - if both sides initiate CONNECT and RESUME primitives simultaneously:

- collision

It occurs if two different process-to-process transport connections try to reserve the same VC. This conflict is handled by the TRA sub-layer, which rejects the reservation for one of the connections. The CON sub-layer then has to RESERVE another circuit.

- contention.

  In this case two different circuits are established for the same process-to-process transport connection. The connection administration entity of the CON sub-layer has to detect contention. It is assumed that the addressing mechanism is sufficient to detect contention. When contention is detected each side has to evaluate whether the outgoing or incoming CONNECT has priority. This evaluation may be based on the asymmetric subscriber addresses, for example. The side with the lower subscriber address has to reject the incoming CONNECT. The other side has to accept the incoming CONNECT.

Collision and contention may also occur between CONNECT - RESUME, RESUME - CONNECT, and RESUME - RESUME primitives. These cases are handled in an analogue manner, giving RESUME priority over CONNECT.

DISCONNECT is the interface primitive responsible for terminating of the end-process to end-process transport connection. It uses the FREE primitive or the TRA sub-layer with "disconnect" as parameter.

5.4 Break Recovery

Recovery after network initiated clears is one of the most important features for a reliable transport service based on VC's /11/,/12/. In the proposed transport connection protocol this is provided explicitly or implicitly ("automatic break recovery"). To recover from a network initiated clear a RESUME function has to be performed. Whether the RESUME is accepted or rejected the states on both sides are again synchronized.

The RESUME function can be performed either by the user of the CON service (explicit recovery of broken circuit) or implicitly by a service called "break recovery". The second case gives the justification to introduce a sub-layer above the CON sub-layer providing recovery after a VC is "broken".

6. CONCLUSIONS

In providing high reliable and cost effective transport service for computer based mail services careful design is necessary to fulfill the requirements for functions concerning transport connection management. The logical difference between network connections and transport connections (as proposed in the Reference Model /7/) is the key to provide transport connections of higher reliability than network connections. The service of temporary "release" and "resume" network connections, while transport connections exist further, is a by-product of this design of a transport layer. The provision of a highly reliable transport connection management protocol, which is independent of the reliability class selected through the data transfer phase, gives both the advantage of maintaining a transport connection only while processes <u>are</u> communicating, and the advantage of releasing the transmission resources during the lifetime of a transport connection, while processes <u>are</u> temporarily <u>not</u> communicating.

7. ACKNOWLEDGEMENTS

This document is based on the assumptions made within the "PIX-ML-Implementierungs-Gruppe". The work is the continuation of the Message Link Protocol effort started in 1977 by E. Raubold, F. Hertweck and F. Vogt. Further work was done on this topic by G. Bochmann and F. Vogt (in 1978). The work on the specification of the services, the formalized specification and the analysis of the protocol done by G. Bochmann in 1979 (prepared under contract for the Gesellschaft für Mathematik und Datenverarbeitung in Bonn) gave an additional understanding of the protocol. Especially the work on the formal analysis of the protocol done by G. Bochmann led to further subdivision of the transport connection protocol and discovered that the initial protocol was not completely reliable. I would like to thank G. Bochmann for his exceptional work on this topic.

8. FOOTNOTES

[1] HMINET II is a local area (within the Hahn-Meitner-Institut) packet-switching network with X.25 interfaces to their subscribers. It is sponsored by the Bundesminister für Forschung und Technologie of Germany F.R.

[2] DATEX-P is the public packet switching network of the Deutsche Bundespost (Germany F.R.). The first node (BERPEX) was installed in Berlin in 1978 as part or the BERNET project.

9. REFERENCES

/1/ Hertweck,F., Raubold,E., and Vogt, F., X.25 based process-to-process communication, Computer Networks, Oct. 1978

/2/ Hertweck, F., Raubold, E., and Vogt, F., The ML Protocol Description, PIX/HLP/TAG/78/01

/3/ Bochmann, G.v. and Vogt, F., Message Link Protocol, Functional Specification, ACM-SIGCOMM, Computer Communication Review, (Vol. 9, No. 2)

/4/ Bochmann, G.v., Specification of the Services provided by the MLP Département d'informatique et de recherche opérationelle, Université de Montréal, April 1979

/5/ Bochmann, G.v., Formalized Specification of the MLP, Département d'informatique et de recherche opérationelle, Université de Montréal, June 1979

/6/ Bochmann, G.v., Analysis of the MLP, Département d'informatique et de recherche opérationelle, Université de Montréal, June 1979

/7/ ISO/TC97/SC16/N227, Reference Model of Open Systems Interconnection, Version 4, August 1979

/8/ Rybczynski, A.M. and Weir, D.F., DATAPAC X.25 service characteristics, Proc. Fifth Data Comm. Symp., ACM/IEEE 1977, pp.4-50 to 4-57

/9/ C.C.I.T.T., Provisional Recommendation X.3, X.25, X.28, X.29 on packet switched data transmission services, Geneva 1978, ISBN 92-61-00591-8

/10/ Sunshine, C.A. and Dalal, Y.K., Connection Management in Transport Protocols, Computer Networks 2 (1978) 454-473

/11/ Study of efficient methods for achieving reliable access to packet switched networks project carried out for the Compt. Comm. Group, Canada, and Master's project, Département d'IRO, Université de Montréal

/12/ Chung, R.J. and Rybczynski, A.M., Alternatives for providing highly reliable access to X.25 networks, submitted to NCC 79, New York

/13/ Vogt, F. et al., Specification of a transport and session layer protocol, ISO/TC97/SC16/WG6 Gaithersburg 10, June 1980

/14/ ECMA Transport Protocol, ECMA/TC24/80/67, ECMA/TC23/80/111, July 1980

/15/ Schicker, P., The computer based mail environment - An Overview Bell-Northern Research Ltd., Dept. 3D23, P.O.Box 3511, Station C, Ottawa, Ontario, Canada

/16/ Pickens, J.R., Functional distribution of computer based messaging systems, Proceedings of the Sixth Data Communication Symposium, Nov. 27-29, 1979, Pacific Grove, California, pp.8-17

/17/ Bochmann, G.v. and Pickens, J.R., Formal specification of a message transport system SRI International, Telecommunications Sciences Center, Working Paper No. SRI/TSC/80-1

/18/ Rybczynski, A.M., X.25 Interface and End-to-End Virtual Circuit Service Characteristics, IEEE Transactions on Communications, Vol. Com-28, No. 4, April 1980

/19/ Rybczynski, A.M. and Palframan, J.D., A Common X.25 Interface to Public Data Networks, North-Holland Publishing Company, Computer Networks 4 (1980) 97-110

TELETEX

Chairman - I. Cunningham

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

PROTOCOL ARCHITECTURE AND ITS APPLICATION FOR CCITT STANDARDIZED SERVICES

Wilhelm Staudinger

Deutsche Bundespost Fernmeldetechnisches Zentralamt
D-6100 Darmstadt

Federal Republic of Germany

Basing on the work of ISO, CCITT is studying a Reference Model for data communication protocol architecture.

The model is intended to apply to any user applications and to all network types. The model not only applies to customer defined services but also to CCITT regulated services like Teletex, facsimile, and Interactive Videotex.

CCITT has succeeded in defining a network independent transport protocol, initially intended to be applied for the Teletex service but already anticipating much broader application for other kinds of services.

REGULATED SERVICES VERSUS CUSTOMER-DEFINED APPLICATIONS

Public telecommunication networks not only support the "regulated" services but also carry a high traffic volume arising from "customer-defined" applications.

"Regulated services" are comprehensively standardized by CCITT, characterized by

- full end-to-end compatibility, guaranteed by the telecommunication administration [1], of CCITT standardized terminals;
- CCITT standardized test procedures and rules of operation to maintain the service quality criteria, including provision against misuse;
- administrative aids like subscriber directory and information inquiry services.

Telephony, telex, and Teletex are examples of regulated services.

"Customer-defined" applications are tailored to a customer's particular needs. A particular customer may need other or much more sophisticated services than generally needed by the majority of customers and therefore provided by the regulated services. He may define his own procedures and, observing the interface conditions, use the public telecommunication networks as the transport medium. The transport medium can certainly not provide compatibility

between different non-regulated applications; it merely provides a transmission path between sources and sinks, unless conversion facilities are made available. This is true, in principle, for all types of network, may it be a telephone network, a circuit switching network or a packet switching network. Only the degree of access compatibility is different.

Examples for the usage of public telecommunication networks are data transmission via the telephone network (V.-series CCITT Recommendations) and data transmission via the public circuit or packet switching data networks (X.-series CCITT Recommendations).

PROLIFERATION OF CUSTOMER-DEFINED APPLICATIONS

Customers have been appreciating that a variety of services, facilities and modes of operation are available on public telecommunication networks, that they can choose those types of networks which best meet their technical as well as economical requirements, and that they can establish according system architectures. This freedom has, however, led to a variety of incompatible system architectures which makes it difficult to move from one network to another, or to change liaison with one particular manufacturer who has so far designed and maintained the user's communication system, or to communicate with dissimilar customer systems.

THE NEED FOR A STANDARDIZED SYSTEM ARCHITECTURE FOR CUSTOMER-DEFINED APPLICATIONS

The shortcomings which are identified in the present area of customer-defined data communication systems, i.e. incompatibility of different systems, network and manufacturer dependence, and difficulties regarding step-by-step innovation in existing systems, urge for a generally applicable system architecture.

The first step in achieving this aim is a methodical representation of existing network and terminal functions.

A further step is to try to accommodate conflicting requirements and to overcome network constraints with the aid of a universally applicable logical system structure.

During the 1976-1980 Study Period of CCITT an urgent need had been recognized to try to elaborate a system architecture which will make future communication systems transparent and open for easy interconnection of any kind. CCITT Study Group VII being responsible for new data networks had nominated a Special Rapporteur with the mandate to formulate a Question concerning the system architecture. At the Final Meeting of Study Group VII (7-15 February 1980) the text of a new Question was adopted (Table 1), revised in subsequent Special Rapporteur's meetings, and finally approved by the CCITT Plenary Assembly in November 1980.

A REFERENCE MODEL FOR PUBLIC DATA NETWORK APPLICATIONS

The CCITT Study Group VII Special Rapporteur did not confine his work just to the formulation of a new Question but used the time already for efficient studies. His efforts were supported by the generally recognized need to ccordinate the emerging development of data communication systems at the earliest possible stage.

The CCITT

considering that

a) Recommendations X.1 and X.2 specify current Public Data Network services and facilities;

b) Administrations* in many countries are planning to establish telecommunication services which will exploit the Public Data Networks now existing or due to be available in the near future;

c) these services may be carried on different types of networks;

d) users of these services need to communicate with each other irrespective of the diversity of interconnected networks;

e) methodical representation of network services will further promote the effective and efficient use of Public Data Networks;

f) accommodation of conflicting service requirements and network constraints can be effected through the analysis of service and network functions. This analysis should lead to a universally applicable logical structure which may be used in the development of compatible service, interface and procedure definitions;

g) this structure should as far as possible accommodate existing Recommendations to allow the steady evolution of networks providing the new services;

h) close collaboration and liaison with other bodies studying reference models is desirable to ensure the widest possible applicability of resulting Recommendations;

i) work is going on in other Study Groups with regard to standardizing a network independent transport service optimized for and dedicated to the regulated Teletex and facsimile services;

1. What structure should be recommended for a reference model for Public Data Network applications to assist in the specification, design, development and application of present and future services and facilities?

2. How should the recommended reference model be employed to facilitate the further development of Public Data Networks, and services which may use Public Data Networks, to meet user and Administration* requirements?

3. What additional Recommendations are required to satisfy the functions described in the model?

Table 1: Draft New CCITT Question on a Reference Model

The International Organization for Standardization ISO had initiated the study of an architectural model (TC97/SC16); it is obvious that the CCITT groups took the results gained in ISO as a basis for their work. Several meetings had been organized. They led within a relatively short time to a Reference Model which is intended to apply to any user application in a data terminal equipment, or a public data network which requires logical processes, as well as to all types of CCITT services, e.g. circuit and packet switched services.

The model is established in the form of a partitioned set of functional units, where each partition provides a unique function.

The partitioning into layers shall permit new technologies to be introduced without affecting the whole architecture but only those layers which are directly concerned; e.g. a move from network type A to network type B should only affect those layers which provide the network dependent services.

A schematic presentation of the Layered Reference Model is given in Table 2. The upper 3 layers of the 7-layer model describe the particular application and its presentation as well as the coordination of interactions.

Layer Number	Function	Examples
7	Application	Data Base Time Sharing Electronic Funds Transfer Order Entry
6	Presentation Control	Data Structure Formats Virtual Terminal Protocol File Transfer Protocol
5	Session Control	Session Management
4	Transport End-to-End	Network Independent Interface
3	Network Control	X.25 - Level 3
2	Link Control	X.25 - Level 2
1	Physical Control	X.21

Table 2: Structure of a Reference Model

The lower 3 layers adhere to the network characteristics, in other words, they relate to interface recommendations like X.21 or X.25.

Major importance is attributed to the transport layer between the upper and the lower 3 layers. The transport layer is intended to relieve those layers which are related to the application, its presentation and coordination, of any concern with the detailed way in which the type of data transfer requested by the higher layers is achieved. The transport layer may also add higher sophistication to the underlying network, if required, e.g. multiplexing functions, error recovery procedures, etc.

The definition of the Layered Model aims also at a complete network independence of the application and its administration. This is specifically discussed in the next chapter.

In contrast to the closed groups of application which, at the present time, populate the telecommunication networks, those future systems which conform to the model and employ related CCITT Recommendations thus allowing "open" communication which is not limited by certain system characteristics, may be referred to as "open systems".

NETWORK INDEPENDENCE

One of the major benefits of the application of the layered model is intended to be network independence. On the one hand customers appreciate that different networks with different service characteristics and different tariff structures are at their disposal; on the other hand the manufacturers suffer from the need to adapt their products to the various network characteristics which, in many cases, differ basically from each other. For this reason it would be highly beneficial to achieve a network independence of data communication systems.

PRESENT STATE OF DISCUSSIONS

CCITT Study Group VIII [2)] takes full responsibility in the study of the characteristics of Teletex terminals, Study Group XIV [2)] studies facsimile terminals, and Study Group VII (Data Communication Networks) has taken up the study of the Reference Model.

The architecture of the Layered Model has been established, generally explained and been offered to CCITT in form of a contribution. It has been discussed in special groups within Study Group VII, but certainly not yet fully discussed or even approved by the whole body of Study Group VII. It has to be recognized that all implications which may be expected for networks as well as for terminal equipment could not yet be studied in all detail. The authors of the CCITT contribution themselves have admitted that "it is difficult to select an optimum model for all services, therefore, it is expected that this model will be modified as new service requirements or model deficiencies are identified".

As far as the network independent transport service is concerned it has been recognized that network replacement or interconnection may be achieved easily only if even networks provide the same elements of service. This is, however, not the case in present networks; the services e.g. offered by telephone, circuit or packet switching networks are far from providing one-to-one correspondence.

Two approaches to achieve network independence were proposed, i.e.

to add functions to existing standards to raise them all to one single network service standard, which obviously is the highest one offered by all candidates, or to modify existing standards so that they inherently provide the generally agreed network service.

It is highly appreciated that the Special Rapporteur's Group of CCITT Study Group VII has achieved fundamental results which guided largely the work on the Teletex protocols. Even if these results could not be finalized to such an extent that they could be generally accepted, they were considered to be of such a good value that they should be annexed to the new Question for CCITT Study Group VII "Structure for and use of a reference model for Public Data Network applications". There are two annexes which are entitled "General explanation of the reference model for Public Data Network applications" and "Elements of a network independent transport service".

The entity of the new Question will not only form the basis of the Study Group VII work in the new study period, as approved by the CCITT Plenary in November 1980, but has already given fruitful input to the Teletex work of Study Group VIII. In return, the results gained by Study Group VIII will certainly influence the future work on the more general basis in Study Group VII.

APPLICATION OF THE REFERENCE MODEL FOR CCITT REGULATED SERVICES

Concern has been expressed by CCITT as well as by ISO that all applications including new text communication services like Teletex, digital facsimile, Interactive Videotex, etc., should be covered by the Reference Model.

In order to coordinate the studies in the different Study Groups as far as the network independent transport service is concerned, a special "Task Group" consisting of members from all three Study Groups was installed to speed up the work on common protocols for Teletex and facsimile. The work of the Task Group confirmed the general intent to follow the guidelines of the Reference Model as closely as possible, but also made apparent that it is extremely difficult to satisfy the needs of circuit switching networks and the needs of packet switching networks in a balanced manner, in particular in view of flow control, error procedures and multiplexing functions.

Nevertheless, the highly concentrated efforts of the Task Group and a Study Group VIII adhoc group of protocol experts resulted in a draft Recommendation entitled "Network Independent Basic Transport Service for Teletex". The defined transport service protocols are applicable to all types of basic Teletex terminals whether they are meant to be connected to circuit or packet switching data networks or to the general switched telephone networks (see Table 3).

The general structure of the transport service including the funcional blocks of layers 1, 2 and 3 is presented in Fig. 1. The diagram shows clearly that the basic characteristics of the different network types have been retained but certain functions added in the case of circuit switching data and telephone networks, and certain functions modified or specifically determined in the case of packet switching networks.

The CCITT considering that

- The Teletex Service will be introduced in different types of network, i.e.

  i) Circuit Switched Public Data Network (CDSN)
  ii) Packet Switched Public Data Network (PSDN)
  iii) General Switched Telephone Network (GSTN)

- There is a need for international interworking between Teletex terminals connected to different types of networks.

Unanimously declares that:

- This Recommendation defines the Network Independent Basic Transport Service (NITS) applicable to Teletex terminals connected to the above mentioned types of network in terms of the

- Transport Services provided to the higher layer. The Transport Services are provided by the transport layer (layer 4) in association with the underlying services provided by the supporting layers 1-3.

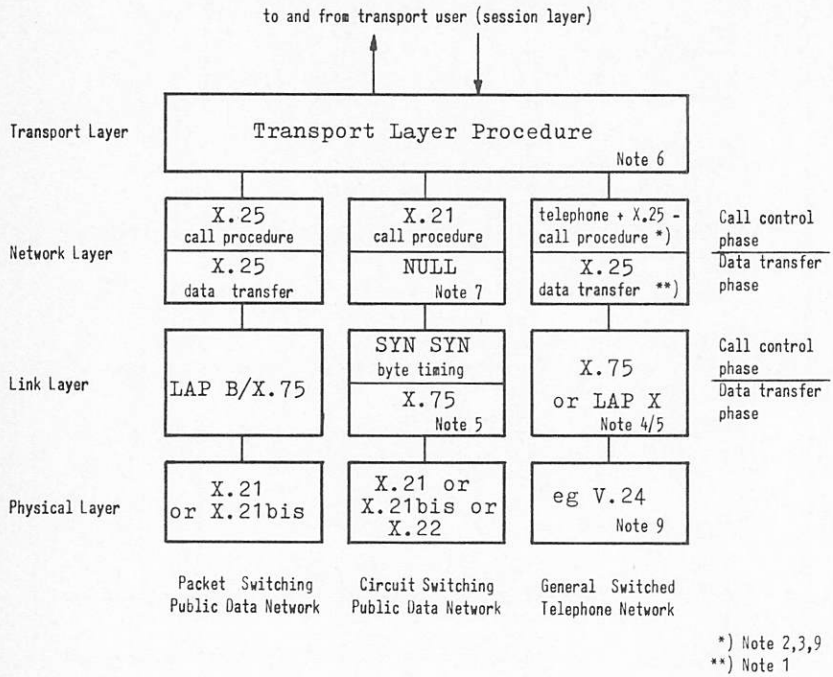
- Transport layer procedure

Table 3: Introduction to CCITT Recommendation S.70

This draft Recommendation was finalized at the Final Meeting of Study Group VIII in Montreal, June 1980 and was approved by the CCITT Plenary Assembly in November 1980. It will be published as CCITT Recommendation S.70.

Recommendation S.70 does not yet contain a specification of LAP X, the link access protocol for half duplex operation in the general switched telephone network. Study Group VIII, at the Montreal meeting, appointed therefore a Special Rapporteur who succeeded - together with his collaborateurs - in establishing a draft recommendation on "LAP B extended for half-duplex physical level service". The CCITT Plenary Assembly authorized the CCITT Secretariat to distribute this draft recommendation to the member countries for letter ballot. At the present time (January 1981) the letter ballot procedure is in process.

Several PTT administrations are working out specifications for their national implementation of Teletex. These specifications take full account of CCITT Recommendation S.70, as in the case of the specification of the Deutsche Bundespost for the Teletex terminal equipment which shall operate in the circuit switched 2400 bit/s data network. This specification is already finalized.



Notes:

1. The X.25 network layer procedure is introduced to ease inter-working with PSDN's.
2. The establishing of the network connection id performed in two stage selection, first using normal telephone procedures and second using X.25 call control procedures.
3. For terminals connected to GSTN accessing PSDN, the procedures in note 2 apply.
4. LAP X is a half-duplex link access procedure, based on recommendation X.75 for single link operation.
5. The link level procedures are in accordance with X.75 for single link operation and in this respect Recommendation X.75 is to be regarded as the reference specification of this protocol.
6. In case of interworking between Teletex Terminals connected to different types of networks (i.e. CSDN), PSDN, GSTN) this Transport Layer Procedure is executed peer-to-peer between the communicating Teletex Terminals.
7. For terminals connected to CSDNs, no function is needed in the network layer in the data transfer phase. However, in order to facilitate interworking with PSDNs a minimum network layer is introduced.
8. The modem may also be integrated within the terminal and in such cases recommendation V.24 need not apply.
9. For automatic calling and/or answering recommendation V.25 may be applicable.

Fig. 1: General Structure of the Transport Service

As far as digital facsimile is concerned there is a high degree of similarity with Teletex: facsimile terminals are, in the first instance, single call terminals for terminal to terminal communication, in the same way as single call Teletex terminals intended to be used for the basic Teletex service as defined by CCITT Study Group I (Fig. 2). It should not cause any difficulty to adapt for new digital facsimile services the same protocol system as for Teletex.

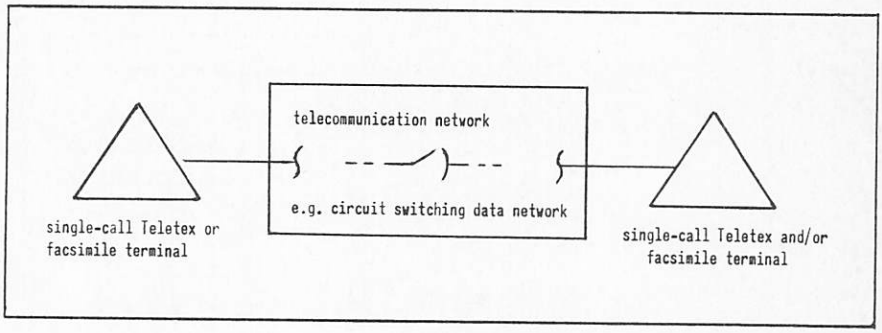

telecommunication network
e.g. circuit switching data network
single-call Teletex or facsimile terminal
single-call Teletex and/or facsimile terminal

Fig. 2: Terminal-to-Terminal Communication of a Basic Teletex/Facsimile Service

Fig. 3 shows that for the Interactive Videotex service two different types of communication are employed:

- interactive communication via the general switched telephone network between a simple single call video terminal operated by a human - usually a home TV set - and the Videotex centres which may combine the function of a host computer and the function of a computer in a distributed computer communication system,

- communication between different Videotex centres as well as communication between Videotex centres and external data bases.

At first sight both types of communication requirement seem to have only little commonality as compared to the terminal to terminal communication of facsimile and Teletex.

The possible need, however, to allow interworking between various kinds of different services like Teletex, facsimile, Interactive Videotex, and the general tendency to avoid proliferation of similar standards makes it highly desirable to examine whether the Teletex transport protocols are also applicable to other services, e.g. Interactive Videotex. In fact, some administrations are already considering to use a transport protocol according to CCITT Recommendation S.70 for their national Interactive Videotex service.

In the Federal Republic of Germany a national working party including representatives from the Deutsche Bundespost, the Association

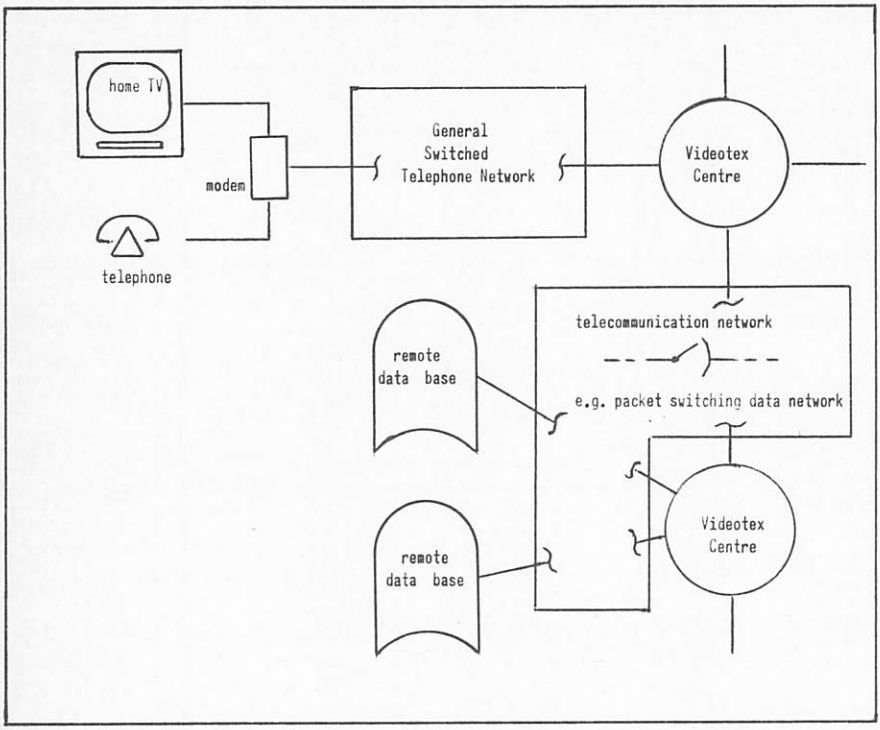


Fig. 3: Communication System of an Interactive Videotex Service

of Computer Centres (VDRZ), the Ministry of the Interior, and major users had taken the initiative to establish a national system of "uniform higher communication protocols (EHKP)".

In April 1980 a complete set of protocols for data communication over the public packet switching network, specified in every detail, including the coding, was at the disposal of all national computer communication projects.

It was very clear that the general acceptance of these protocols could be highly promoted if the expected great volume of Interactive Videotex application would use the EHKP. Therefore, great efforts were made during the definition of the EHKP that also the requirements of Videotex are satisfied. For the sake of early national standardization the Deutsche Bundespost decided to support the EHKP and to use them at least for the Interactive Videotex field trial service, as long as no international standard is available.

Unexpectedly early the scene has changed:
CCITT has approved a transport protocol for telephone, packet

switching, and circuit switching networks, offering the great chance to cover requirements both from customer defined data communication applications as well as from a set of regulated services by one single set of protocols. The Deutsche Bundespost therefore is already evaluating these benefits in view of the regular Interactive Videotex service and so do also some other administrations.

GENERAL APPLICATION OF CCITT RECOMMENDATION S.70

It is obvious that the transport protocol defined for the basic Teletex service as laid down in CCITT Rec. S.70 can be applied much more generally and need not at all be confined to Teletex. The development of this protocol was largely based on ECMA 3) papers which had already found wide acceptance, certainly not in view of a regulated service like Teletex but in view of customer defined data communication applications. Taking into account the high number of Teletex terminals which are forecasted by various telecommunication administrations (e.g. 40 000 terminals in the network of the Deutsche Bundespost by 1985) it can be assumed that a number of data communication applications will follow the Teletex model which - because of its high volume - could play a leading role in the terminal market as well as in the area of network services provided by the telecommunication administrations. This view is supported by the fact that ECMA/TC 24 has already incorporated CCITT Rec. S.70 as the class 0 in the ECMA series of compatible transport protocols.

The big chance is offered now that a compatible set of transport protocols for regulated services as well as for customer defined applications will improve the non-voice telecommunication infrastructure. It is up to those who take a decisive role in national as well as international standardization to derive utmost benefit out of this chance.

FOOTNOTES:

1) In the following the expression "telecommunication administration" is used in such a sense that Recognized Private Operating Agencies are included.

2) At the CCITT Plenary Assembly (Nov. 1980) Study Group VIII and XIV have been merged into a new Study Group VIII "Terminal Equipment for Telematic Services (Teletex, Videotex)"

3) ECMA: European Computer Manufacturers Association

REFERENCES:

1) E. Hummel: The role of the CCITT in the introduction of new telecommunication services. Telecommunication Journal Vol. 47-I/1980.

2) CCITT Study Group VII draft Question: Structure for and use of a reference model for Public Data Network applications.

3) CCITT Recommendation S.70: Network independent basic transport service for Teletex.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

TELETEX AND IT'S PROTOCOLS

G A Routhorn, P A Carruthers

British Telecom Marketing Executive
Product Development Unit
LONDON

The paper describes the functions offered by the TELETEX service and examines the high level protocols necessary for its implementation. It discusses the use of the TELETEX high level protocols for more advanced text communication applications and the benefits this brings in terms of upward compatible terminals.

1 INTRODUCTION /1/

The concept of electronic mail is not new, telegrams, facsimile and telex are all versions of it, each developed using the technology of the time, to meet the user's requirements and each has improved as technology has advanced. The use of the computer has not only improved the existing services but, in its micro-processor form is enabling new and demanding requirements to be met. Equipment known as "editing typewriters" or, alternatively, "word processors" is rapidly entering the office environment because of the improved text preparation and manipulation facilities it offers compared with the simple typewriter. Economic and efficiency considerations as well as an indifferent Postal Service in some countries lead to the need for this type of equipment to be able to communicate electrically. However, the products of different manufacturers are often incompatible and unable to inter-communicate. This situation, which in the past restricted the growth of facsimile, clearly demonstrates the need for a standardised text-communication service.

This paper describes the facilities offered by the Teletex service and examines the higher level protocols which have been defined to enable the service to function. In addition, it discusses the use of the Teletex protocols as a more general protocol for other application.

2 WHAT IS TELETEX?

In 1976 CCITT began to study the technical and service requirements for a new text communication service which would offer more sophisticated features than the existing telex service. The name chosen for this service was Teletex and 5 draft recommendations, have been ratified at the 1980 Plenary Assembly. These Recommendations are:-

1 Service definition /2/
2 Technical requirements of terminal /3/
3 Character set and coding /4/
4 The Transport Service /5/
5 End-end protocol /6/

The Recommendations are designed to leave as much scope as possible for the introduction of new facilities which may be required later.

2.1 THE TELETEX SERVICE

The service recommendation for Teletex fixes the rules to be followed by administrations and operating agencies in the provision of an international service. It is intended that subscribers will be able to exchange correspondence on an automatic basis between electronic memories. Figure 1 illustrates a conceptual Teletex terminal.

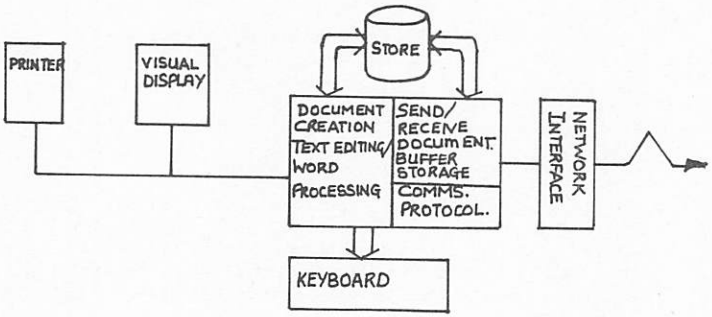


FIGURE 1 :CONCEPTUAL TELETEXT TERMINAL

A simple terminal of the communicating office typewriter type is envisaged as the basic terminal although more sophisticated terminals will be subjects for the next stage of standardisation, for example, a unit incorporating a cluster of work stations. The important feature is that users should be able to prepare their correspondence and enter it into the memory for the system to deliver it. This is analogous to posting a letter and relying on the postal service to deliver it. The provision of a memory at the receiving station ensures that a user preparing correspondence need not be interrupted by incoming traffic.

The 2400 bits per second recommended transmission rate for international working is much greater than the 50 bits per second used in the Telex Service. This rate has been chosen in order to allow a high grade of service to be given and the service recommendation lays down requirements which should mean that delays to traffic caused by calling terminals being engaged will be small.

However, initially at least, administrations may elect to use a lower rate within their own country in order to keep down costs.

The service recommendation demands a unique identification signal for each terminal participating in the service and administrations are required to publish directories which include an alphabetic list of subscribers with their business descriptions and the identification signals which have been allocated to them.

Interworking with Telex is an essential feature of the basic service and this should give users a useful communication service even before a significant number of Teletex terminals has built up.

2.2 TERMINALS

A technical recommendation lays down the requirements for a Teletex terminal. In drafting this recommendation a guiding principle has been that, as far as practicable, users of the service should have as much freedom in the way

they prepare correspondence as they have when using mail services. This obviously gives rise to problems because, when preparing a document for mail transmission, the sender is in control of the reproduction and has freedom to choose type font, colour of paper, colour of type, size of paper etc. In the case of Teletex however reproduction is done remotely from the sender and, in order for the received document to have the same content, layout and format as that sent, all usable options have to be standardised together with the protocols needed for controlling their use. In practice it has been necessary to define a basic service incorporating features which must be provided by all terminals. Additional optional facilities are also standardised but will not necessarily be available on all terminals.

The basic terminal is designed to work on a page basis with the common usable area of A4 and American Standard paper as the basic page. The paper can be used with the typing parallel to either the short or long dimension. In the latter case it is referred to as "A4 Landscape" or more commonly "A4L" format. To feed paper automatically in either format upon demand presents mechanical problems in the design of equipment and the terminals designed so far have avoided the problem either by adopting wholly manual insertion of paper or by automatic insertion of paper in the A4 mode but manual in the A4L mode. Either method is quite acceptable even at the receiver because traffic is stored electrically when received and the printing need not take place until an operator is in attendance. In fact printing of the received text is not mandatory for other forms of display are permitted.

Because of the international nature of Teletex the basic character repertoire is much greater than that normally provided by a typewriter. The repertorire comprises some 52 small and capital letters of the Latin Alphabet, the characters formed from these by the addition of diacritical marks and also a range of numerals, punctuation marks, currency signs etc. The total is in excess of 300 and a Teletex receiver must be capable of displaying or printing all of them in an adequate manner. It is accepted that to obtain an accurate reproduction of character form it may be necessary to change the print element for one designed specifically for a particular sub-set of the repertoire. For example, whilst a British user would be able to receive text prepared in the Icelandic language, his printer would normally be optimised for English so that, if he regularly received traffic from Iceland, he might wish also to have a print element optimised for Icelandic which he could fit prior to printing out Icelandic traffic.

Text processing facilities, which are usually local functions, range from the simpler character and word editing to sentence and paragraph manipulations. Standard letters may be stored and accessed as required.

The Teletex service provides a standardised means for the communication of text and it offers both basic and optional facilities.

The requirements of different businesses gives rise to different terminal configuration ranging from the simple simple stand-alone type to clusters of terminals, sharing the resources of a central controller which also gives access to the Teletex services. In addition the larger businesses may require larger PABX type systems.

Clearly the Teletex terminals will come in a variety of forms depending upon the mix of text processing, communications and customers' business requirements.

3 TELETEX PROTOCOLS

To exploit the potential of teletex as an electronic mail media, it was essential that its communications protocols were standardised on an international

basis. The sophistication of the protocol required to allow the error free store to store transfer of text documents was an order of magnitude greater than that required for telex.

To aid in the design of the protocol use was made of the ISO open system model /7/ see Figure 2. This model, initially conceived for design of computer communication networks, sub-divides the complex communication into more easily understandable modules or layers and allows "open" communication between equipments connected on various types of communication networks. Each layer provides services to the next higher layer by making use of the services of the layer immediately below. The layer in one station logically communicates with the same layer in another station using an agreed set of rules or protocols. The interface between adjacent layers is not standardised leaving terminal manufacturers free to design adjacent layer interface in their own way allowing maximum flexibility. Within the model, a clear distinction can be made between the levels providing for the error free transportation of blocks of data from one station to another, known as the transport station, and the higher service orientated levels which should be completely network independent.

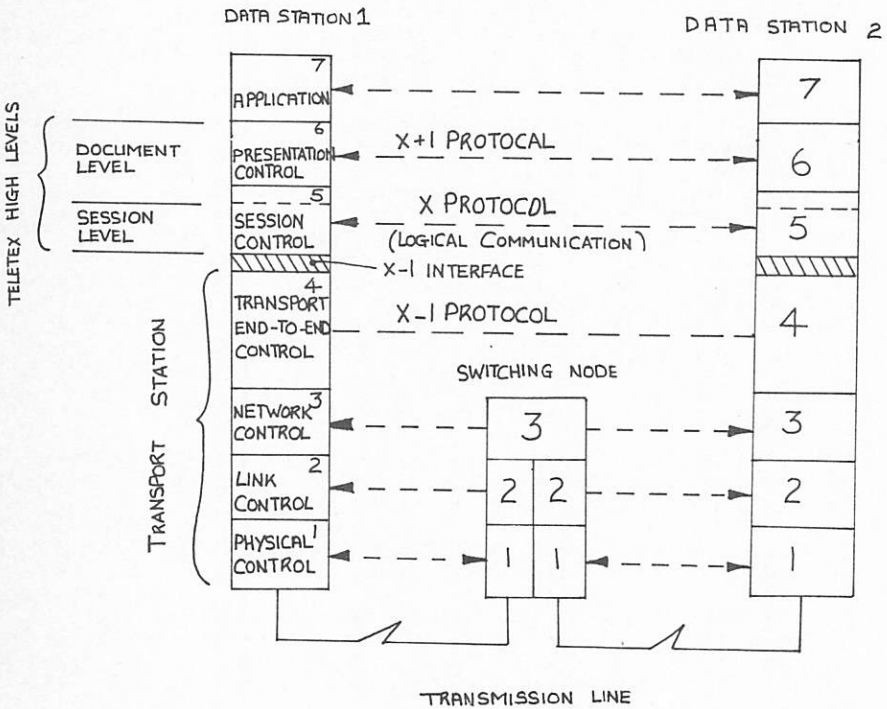


FIGURE 2 :THE I.S.O. ARCHITECTURAL MODEL

3.1 TRANSPORT SERVICE

A network independent transport service has been developed for use with Teletex. The transport service has end to end significance and provides for the error free delivery of information from end user to end user posing no restriction on the content of the information fields. Transport service users can select from the resources available, the performance required to give maximum cost effectiveness for the call.

To achieve this the transport service makes use of the network link and physical layers.

The physical layer provides the physical interface to the communications media (ie V24 to modem). The link level provides for the error protected flow of data using essentially HDLC techniques (retransmission of a frame when an error is detected) and the network layer provides control of call routing. Figure 3 shows the relevant CCITT standards on General Switched Telephone Networks (GSTN), Circuit Switched Data Network (CSDN) and Packet Switched Data Network (PSDN).

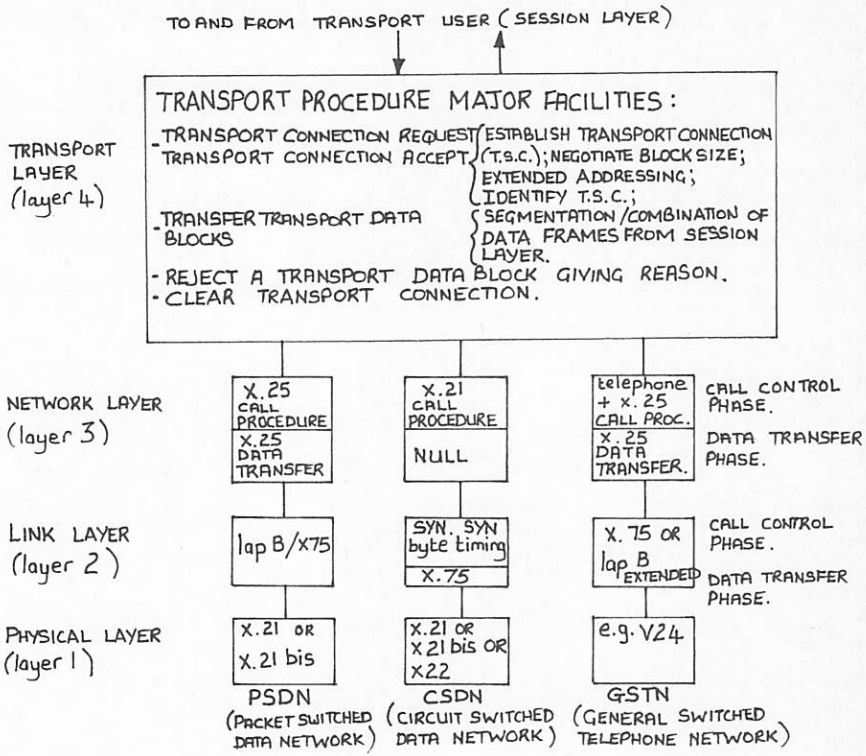

TO AND FROM TRANSPORT USER (SESSION LAYER)
TRANSPORT PROCEDURE MAJOR FACILITIES:
TRANSPORT LAYER (layer 4)
- TRANSPORT CONNECTION REQUEST / TRANSPORT CONNECTION ACCEPT { ESTABLISH TRANSPORT CONNECTION (T.S.C.); NEGOTIATE BLOCK SIZE; EXTENDED ADDRESSING; IDENTIFY T.S.C.;
- TRANSFER TRANSPORT DATA BLOCKS { SEGMENTATION / COMBINATION OF DATA FRAMES FROM SESSION LAYER.
- REJECT A TRANSPORT DATA BLOCK GIVING REASON.
- CLEAR TRANSPORT CONNECTION.
NETWORK LAYER (layer 3)
X.25 CALL PROCEDURE
X.25 DATA TRANSFER
X.21 CALL PROCEDURE
NULL
telephone + x.25 CALL PROC.
X.25 DATA TRANSFER.
CALL CONTROL PHASE.
DATA TRANSFER PHASE.
LINK LAYER (layer 2)
lap B/X75
SYN. SYN byte timing
X.75
X.75 OR lap B EXTENDED
CALL CONTROL PHASE.
DATA TRANSFER PHASE.
PHYSICAL LAYER (layer 1)
X.21 OR X.21 bis
X.21 OR X 21 bis OR X22
e.g. V24
PSDN (PACKET SWITCHED DATA NETWORK)
CSDN (CIRCUIT SWITCHED DATA NETWORK)
GSTN (GENERAL SWITCHED TELEPHONE NETWORK)

FIGURE 3 :TRANSPORT STATION FOR TELETEX ON P.S.D.N., C.S.D.N. and G.S.T.N.

The network independent transport service is an essential ingredient of Teletex and the impetuous provided by Teletex has resulted in its early agreement by the CCITT. It is not discussed further in this paper however as it is equally applicable to any form of data transmission. In the future, enhanced Teletex may make use of more comprehensive classes of transport services.

3.2 HIGH LEVEL PROTOCOLS

For Teletex, 2 levels have been identified above the transport level, the session and document levels. These levels enable the transfer and presentation of documents between end users. The allocation of functions to each level was not easy and became the subject of considerable debate at CCITT.

3.3 STRUCTURE AND CODING OF THE HIGH LEVEL PROTOCOLS

The structure of commands follows the nested header concepts shown in Figure 4, which also illustrates typical command structures.

Each command/response used in the protocol has a header, by which it is identified, followed by a length indicator giving a length of the parameter field which follows. Each parameter has an indentifier followed by a length indicator giving the length of the parameter value.

To aid de-coding, certain parameters of a similar type normally used together can be grouped using a parameter group indicator and associated length indicator.

Session level frames (user information at the transport level) have additional headers added as they pass through the lower levels.

The importance of the ability to extend the protocol was realised and account taken in the coding of commands, responses and parameters. Table 2 shows how the bit allocation of command and response headers relate to the elementary functions of the procedures.

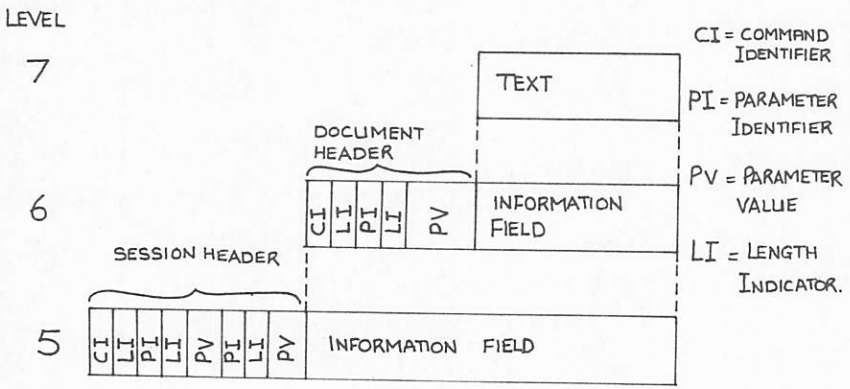


FIGURE 4 :A SAMPLE FRAME STRUCTURE

3.4 SESSION LEVEL PROTOCOL

The session (as defined for Teletex) establishes a logical connection between users to facilitate the document level logical connection. Several documents may be exchanged in a session provided they are all catered for within the options and resources negotiated at the start of the session. In the case of Teletex, all terminals have certain basic capabilities, and so must be capable of exchanging all basic documents.

If transmission of a document is interrupted for any reason, ie corruption, then recovery can be made to the session. (No need to re-establish the transport connection.)

The main functions of the session grouped under initiation, transfer and termination are outlined below. Figure 5 gives one scenario for the exchange of session commands and depicts the relations between, and simultaneous existence of all protocol levels. A state diagram for the session level is shown in Figure 6 and Table 3 lists all the session commands and associated parameters.

3.4.1 Session Initiation

A session is initiated by the sending of the Command Session Start (CSS) and reception of the response, Session Start Positive (RSSP). This initial exchange is a symmetric negotiation, the CSS parameters give the receive capabilities of the initiator and the RSSP parameters reflect the capabilities available at the receiver. The positive response does not necessarily imply compliance by the receiver (it may not even check the parameters of CSS) merely that the command was recognised and the receiver is available to enter a session. The responsibility for proceeding with the session rests with the initiator.

There are 4 aspects to this negotiation:-

i The establishment of session reference information by the exchange of terminal identifiers' date and time, and optionally a reference number. This allows the calling and called parties to exchange identities giving a degree of security to document delivery.

ii The agreement of the protocol/service to be used during the session. Initially it will only be basic Teletex but in the future could be mixed mode, facsimile, enhanced document structure, encryption etc. This negotiation is analogous to the negotiation of virtual terminal classes /8/.

iii Non-basic session capabilities. The only non-basic capability at present is window size but in the future the mechanism can be used for negotiating a two-way simultaneous mode of session, a session suspension capability etc.

iv Non-basic terminal capabilities. This relates to the negotiation of terminal presentation attributes and covers character sets, page formats and other miscellaneous terminal capabilities. It can be argued, quite reasonably, that this aspect is not proper to the session especially as a supplementary mechanism for this purpose is provided at the document level. The presence of these parameters, however, should in no way restrict the future flexibility of the protocol.

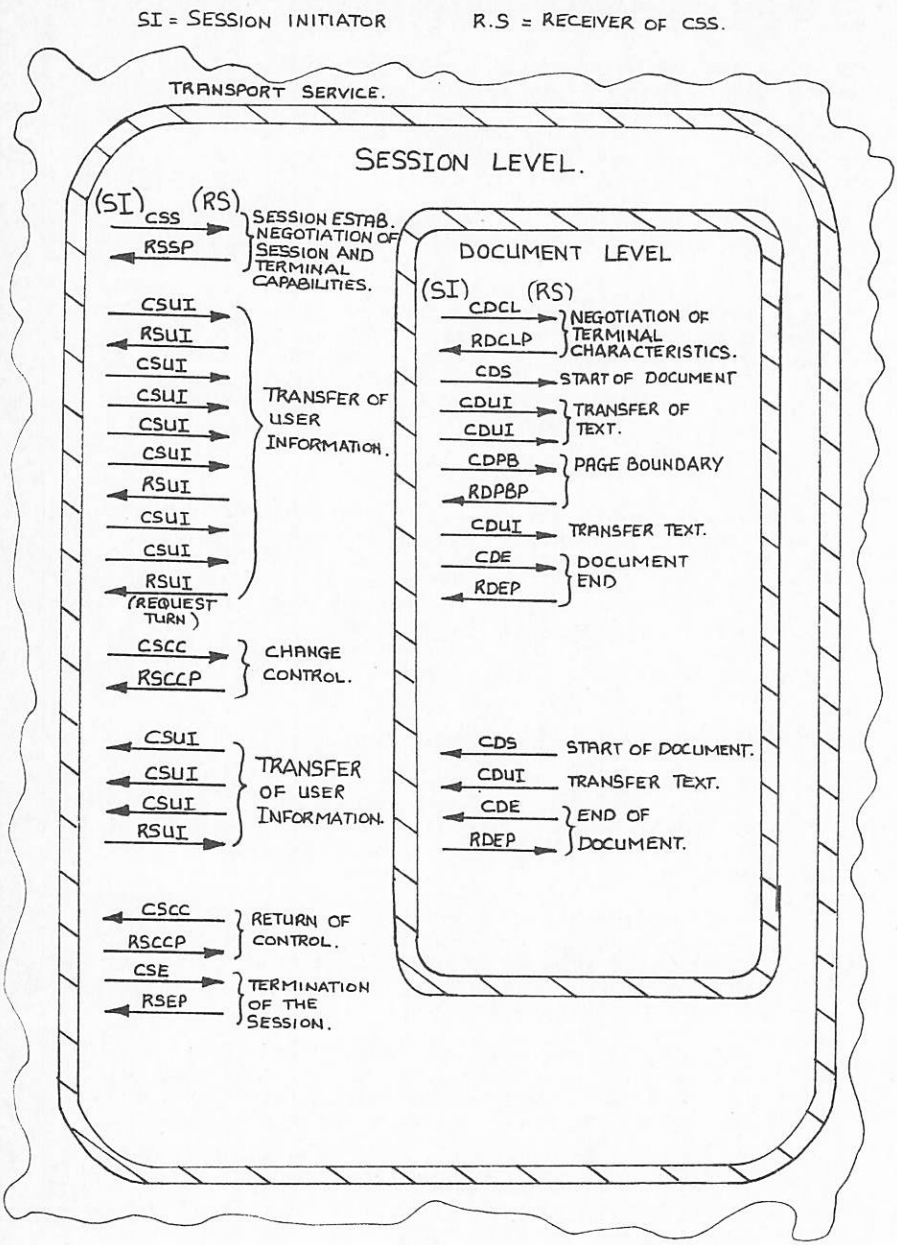
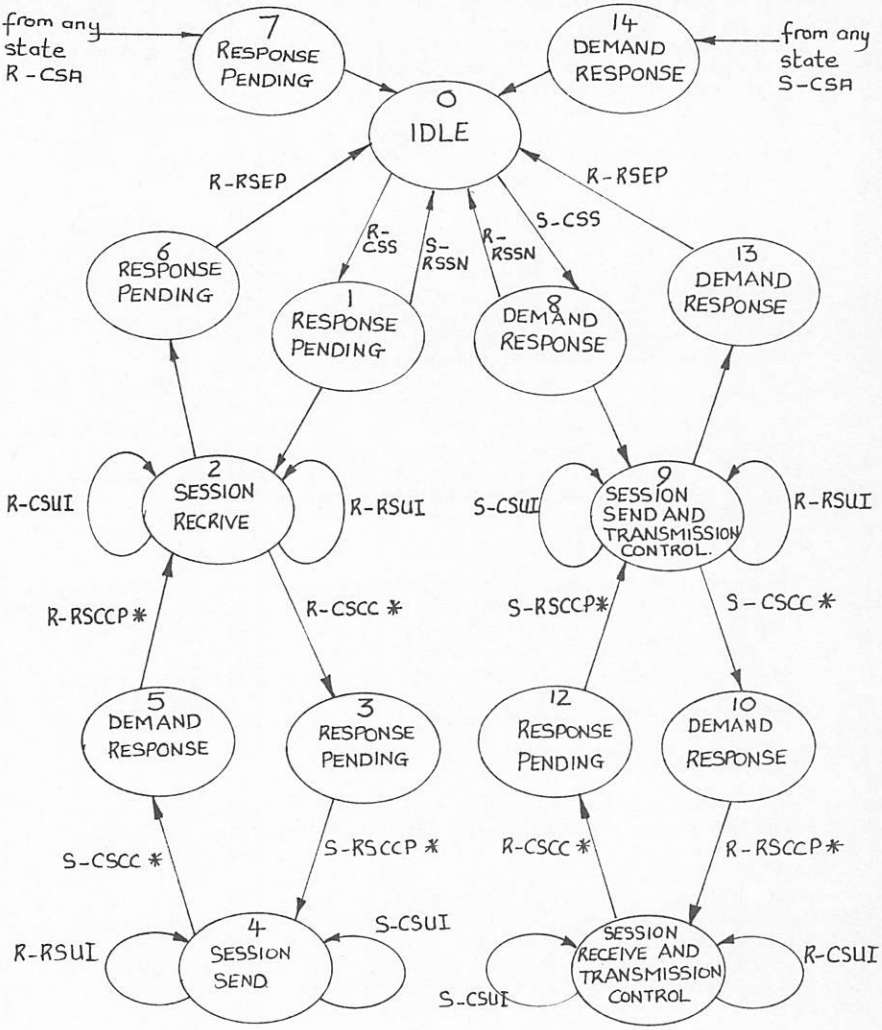


FIGURE 5 :TYPICAL EXCHANGE OF SESSION AND DOCUMENT LEVEL COMMANDS



* THE 'CHANGE CONTROL' COMMANDS DO NOT APPLY TO THE OWC MODE.

FIGURE 6 :STATE TRANSITION DIAGRAM FOR TWA AND OWC SESSION MODES

3.4.2 Transfer Phase

The transfer of information as provided by the transport service is normally full duplex. That is commands and responses may be exchanged simultaneously. The mode of information transfer at the session level, however, can be 'One Way' (OW), Two Way Alternate (TWA), where the session initiator gives the receiver a turn to send documents, or Two Way Simultaneous (TWS) in which case both parties exchange documents simultaneously.

The TWS mode is only provisionally defined and has been identified for further study by the CCITT. TWA is the basic mode. OWA is achieved by default if the session initiator does not send a change control command (CSCC). The session initiator always gets the turn back and is responsible for normal ending of the session. In the event of non-session initiator refusing to cede the turn, the session can always be ended abnormally.

Higher level information (document level) is transferred using the session user information command (CSUI). To allow for the sending of document level responses a response session user information is provided (RSUI). A 'request' turn parameter is provided with RSUI to flag a desire to send a document or regain control.

3.4.3 Termination

Normal session termination is achieved by the session initiator sending a Command Session End (CSE) and receiving a positive response (RSEP). Either terminal may, however, end the session at any time by issuing a Command Session Abort (CSA). A reason for ending the session is given and a parameter value indicates whether the transport connection is to be cleared. The response to this command (RSAP) indicates an orderly termination by the receiver of the (CSA).

3.5 DOCUMENT LEVEL PROTOCOL

This level covers aspects relating to the control of the document transfer and presentation of the document. Central to the design of the Teletex protocols has been the concept of a logical structure based on a document constructed of several pages upon which is presented user text information. This model reflects surprisingly accurately the needs of many forms of text communication and could be expanded in the future to include field and paragraph structures. The blurring of the distinction between dialogue control and document presentation when applying the protocol generally, however, can be confusing. The question of a general and more flexible interpretation is addressed in the next section.

Table 4 gives a list of document level commands and Figures 7 and 5 respectively give state diagram for document level send and receive and one scenario for the exchange of document level commands. The main functions of the document level grouped in the phases of negotiation, initiation, transfer, termination and error recovery are outlined below.

3.5.1 Document Initiation

1 Negotiation

As mentioned in the previous section there is a mechanism at document level for negotiating non-basic terminal capabilities. The negotiation can cover a group of documents or can if desired be carried out before sending each document.

The negotiation is achieved using the Command Document Capability List (CDCL) and the associated positive response (RDCLP).

There are 2 main negotiated attributes, the first refers to non-basic terminal capabilities, (control character sets, graphic character sets, miscellaneous capabilities) and the second receiver storage allocation (in kilo octets). In the first instance the receiver responds by indicating (using its associated parameters);

i that all the requested capabilities are available at the receiver,

ii by giving a list of the requested capabilities which are available at the receiver,

iii by giving the complete list of the receivers non-basic capabilities.

In the second instance, the receiver must include a parameter indicating one of the following:-

i confirmation that the amount of memory requested is available and has been reserved,

ii indication of the available (and reserved) amount of memory in kilo octets,

iii indication that the requested memory capacity cannot be reserved.

iv The receiver cannot estimate memory capacity.

2 Initiation

Having negotiated non-basic terminal capabilities or storage, if required, document transmission is initiated by sending a Command Document Start (CDS), or a Command Document Continue (CDC). No response is given to either of these commands under normal conditions, and both implicitly indicate the start of the page. Parameters associated with CDS are used to indicate:-

i service interworking (ie the document is prepared for forwarding to telex),

ii document type, a normal document would be a Teletex document but other document types have been identified, for instance an operator document which is a priority document for immediate display to the operator, or a control document for communication with intermediate store and forward equipment.

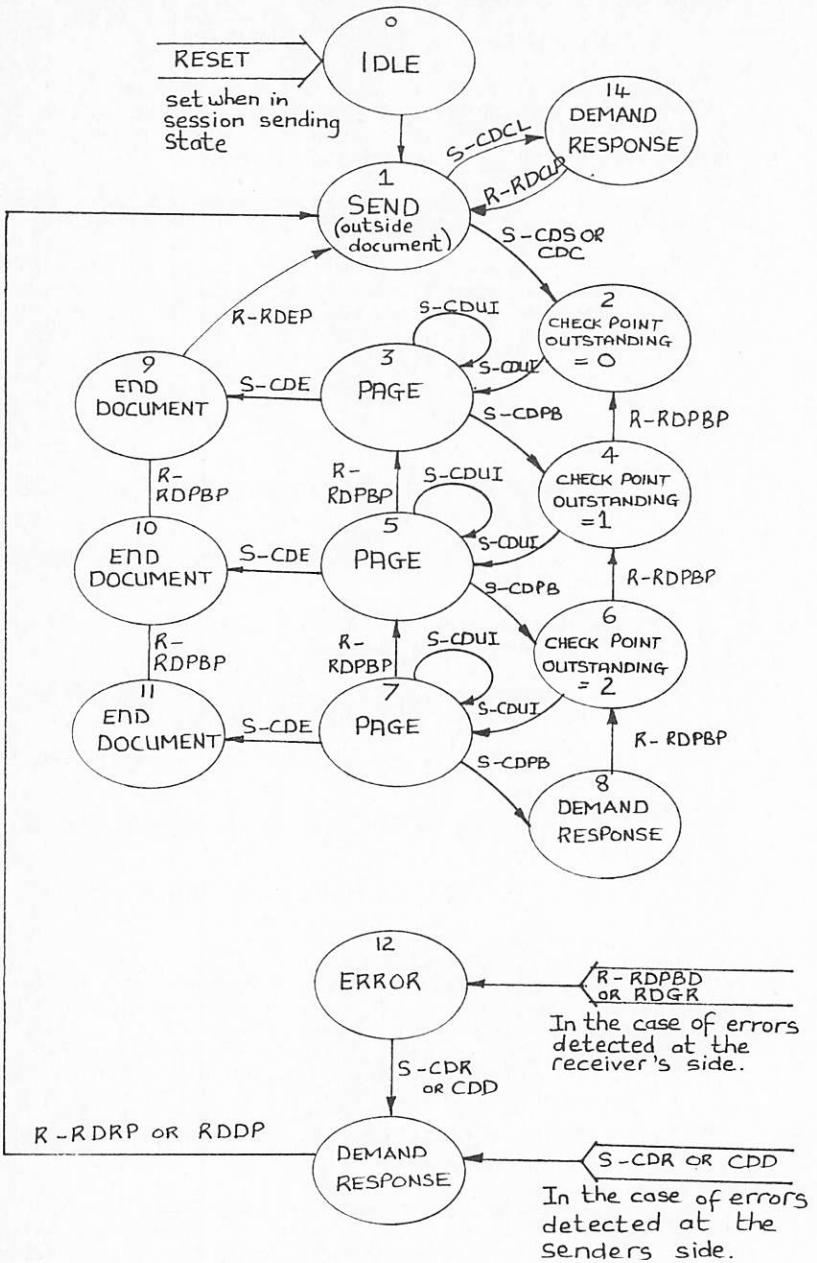


FIGURE 7a: DOCUMENTS STATE DIAGRAM (SENDING PROTOCOL)

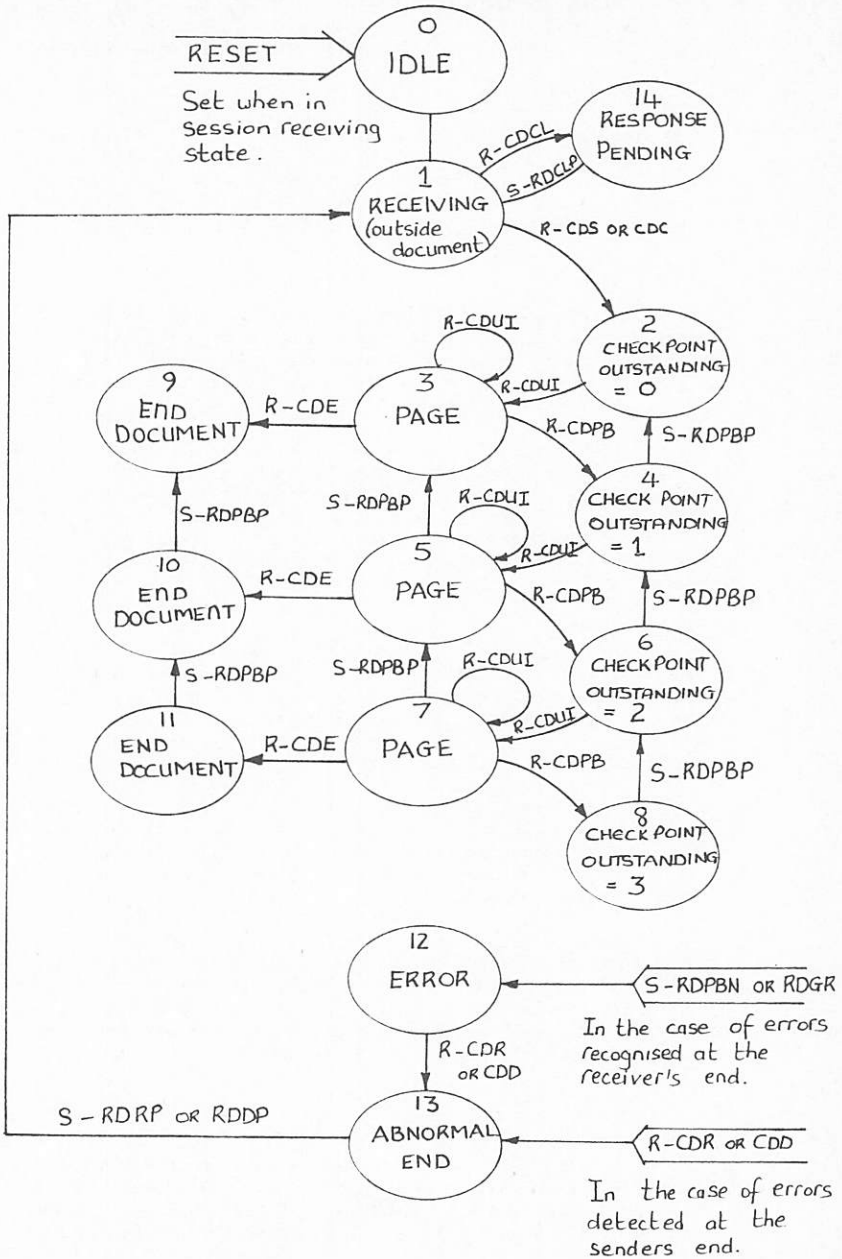


FIGURE 7b: DOCUMENT STATE DIAGRAM (RECEIVING PROTOCOL)

iii document reference number,

iv terminal capabilities required for this document.

(Could be a sub-set of those negotiated using CDCL).

CDC is used to continue the transmission of a partially complete document and in addition to the above parameters must have parameters giving document linking information (document reference number of partially completed document, check point reference number of the last positively acknowledged page).

3.5.2 Transfer

The end user information (Text) is transferred using the Command Document User Information (CDUI). At present no parameters are defined for this command, but as outlined in the next section, parameters could be used to qualify blocks of text (ie mixed facsimile and text information).

Completion of a physical page is indicated by a page boundry command (CDPB) which has an associated parameter, check point reference number, equating in basic Teletex to the physical page number. Completion of a document is indicated using Document End Command (CDE) These commands, as well as conveying user structure information, have additional meaning in terms of dialogue control/error recovery.

3.5.3 Termination

The normal transmission of a document is ended with (CDE) which must be synchronously acknowledged by the response (RDEP). CDE implicitly indicates the last page boundry.

In the event of an error situation it may be necessary to abnormally end the document transmission. This is achieved in two ways. The first is to issue the Command Document Discard (CDD) and await the positive response (RDP). CDD tells the receiver that there is no intention of document linking and that the partially received document can be discarded. The reason is indicated by a parameter (ie local terminal error).

The second is to issue a command document re-synchronisation (CDR), this indicates to the receiver the possibility of linking to the partially completed document and a positive response from the receiver implies that the document up to the last positively acknowledged check point will be stored. The positive response does not imply the receivers capability of linking documents.

3.5.4 Error Recovery

Recovery in the basic Teletex procedures is centred around the CDPB and its responses. Each CDPB has associated with it a check point reference number which must be acknowledged. A window mechanism is defined, (window size can be negotiated at session start and has a default value of 3) in an attempt to preserve information through-

put, but which is not intended to be used for flow control. The usefulness of the window mechanism has been the subject of recent debate.

Once a page (check point) has been acknowledged the sender has no further obligation for storing the page for re-transmission. In the event of a negative page acknowledgement the sender would normally issue a CDR command (acknowledged by CDRP) followed by a CDC quoting the last positively acknowledged check point from which transmission is to be resumed. If the error was severe, the sender may recover by sending a CDD and re-transmitting the whole document. In any event the receiver has the responsibility for discarding duplicate pages. It is interesting to note that CDE has a check point associated with it and can be responded to by RDPBN.

In addition to the errors detected during transmission, which can be corrected by re-transmission, there can be procedural errors. These are indicated, in the case of the receiver by the response, "Response Document General Reject" (RDGR) which flags a desire for re-synchronisation and appends the offending bit pattern as a parameter. Procedural errors detected by the sender result in immediate re-synchronisation (issue of CDR or CDD).

3.6 USE OF THE HIGH LEVEL PROTOCOLS FOR OTHER APPLICATIONS

In defining the protocols, use was made of the CCITT and ISO layered models. This led to considerable debate at CCITT as to which functions were appropriate to which levels and the compromise solution of session and document levels. The biggest shortcoming of the protocol is probably the consequent blurring of the distinction between presentation (user structure) and dialogue control. At the session level CSS negotiates terminal capabilities and at the document level, page boundary and document end commands combine the functions of user structure delimitation and dialogue control.

The actual allocation of function to levels is not so significant, all of the higher level elements can be viewed as belonging to one level. Many Virtual Terminal Protocol(VTP) although thought to reside at the presentation level, have dialogue control elements embodied in them. /9/

The real test of the protocol is whether or not the elements can be used as the basis of protocols for more advanced text transfer and hence allow the development of more complex and yet upwards compatible terminals. We believe the protocol does have this flexibility. The coding of the commands, responses and parameters has been carried out in such a way as to enable the easy distinction of the basic elementary function of the procedures and hence facilitate their use again in other applications. (See Section 3.3.)

Table 1 defines the elementary functions of Document user information, page, document, recovery unit, commitment unit, delivery unit and interaction unit, and shows how they equate to existing document and session level commands. Additional elements are present in the table, the Command Recovery Point (CRP) and Command Recovery Point Re-start (CRPR). These are additional error recovery mechanisms and allow for check points/recovery units within a commitment unit (page). CRP indicates a check point from which recovery may take place and CRPR is used by the source to request re-synchronisation. Re-synchronisation is indicated by the receiver using Response Recovery Point Negative (RRPN).

The likely areas for enhancement of the protocol in the immediate future include mixed mode operation (facsimile and character coded text), the creation of more complex document structures, allowing the transfer of editable documents, and the inclusion of encryption and digital signature options. In the case of encryption proposals have been made which would allow, with the addition of new parameters, the facilities to be incorporated in the existing procedures. Additional document structures, for example field and paragraph could be added by making use of the CDUI with an associated descriptive parameter.

To illustrate the generality of the elements, an example of how they might be applied to transmission of mixed character coded text and facsimile is given. It is not suggested that this would be the optimum means of realising mixed mode operation, the example merely illustrates the flexibility of the procedure. The necessary changes and additions to the elements are given below.

1 The CSS command indicates using the protocol/service identifier that an extended set of protocol elements are required for this session. Should the receiver indicate only basic capabilities the send has the option of falling back to basic operation and transferring normal documents.

2 Special terminal capabilities are negotiated using CDCL (and response).

3 CDS in addition to invoking terminal capabilities indicates the start of a delivery unit. In this instance a delivery unit consists of one physical page.

4 CDPB indicates a commitment unit of 5K octets and has no user structure meaning. With facsimile a physical page could represent a large amount of data, too large an amount for a commitment unit.

5 To allow re-synchronization and ensure adequate through-put, a recovery unit is employed (ie CRP inserted every 1K octets). There is no need for a window mechanism on CDPB.

6 The CDUI is used for transferring document user information. This command now has a parameter used for indicating the type of information (ie facsimile/character coded text) and the area on the physical page which it occupies.

7 The CDE ends the delivery unit and in this instance a physical page also.

4 CONCLUSION

The Teletex protocol has a logical structure defined in terms which are related to its use for text communication.

The coding of the elements of procedure have been effected such that they may be interpreted in terms of elementary functions which relate quite closely to the ISO model. Providing that the coding is implemented as defined, the layer relationship of the elements may be viewed in either text communication or model terms while the functions are preserved.

The approach taken has given rise to a protocol which is flexible and capable of future enhancement. This protocol is the first internationally agreed standard for text communication, it is widely supported and will be subject to refinements

and further enhancement. It is our contention that the Teletex protocol is a strong candidate for a generalised protocol having wider application both in CCITT and other areas.

5 ACKNOWLEDGEMENT

The authors wish to thank the Senior Director of Marketing for his kind permission to use the resources of the Marketing Executive of British Telecom in preparing and presenting this paper. The views expressed are those of the authors and do not necessarily represent the policy of British Telecom.

6 REFERENCES

1 A J Bott G A Routhorn
A PAPER ON TELETEX TO BE PUBLISHED IN THE BRITISH TELECOM JOURNAL (Winter Edition)

2 CCITT Draft Recommendation Fx
TELETEX SERVICE

3 CCITT Draft Recommendation Sc
TERMINAL EQUIPMENT FOR USE IN THE TELETEX SERVICE

4 CCITT Draft Recommendation Sf
CHARACTER REPERTOIRE AND CODED CHARACTER SETS FOR THE INTERNATIONAL TELETEX SERVICE

5 CCITT Draft Recommendation Sh
NETWORK INDEPENDENT BASIC TRANSPORT SERVICE FOR TELETEX

6 CCITT Draft Recommendation Sd
CONTROL PROCEDURES FOR THE TELETEX SERVICE

7 ISO/TC 97/SC 16 N 227
REFERENCE MODEL OF OPEN SYSTEMS ARCHITECTURE

8 Najah Naffah
HIGH LEVEL PROTOCOL FOR ALPHA-NUMBERIC DATA ENTRY TERMINALS
Computer Networks 2 (1978) 84-89

9 P Schicker A Duenki
THE VIRTUAL TERMINAL DEFINITION
EIN/ZHR/77/018B September 1977 pp 48

TABLE 1 ELEMENTARY FUNCTIONS OF THE PROCEDURE

(a) USER STRUCTURE ELEMENTS FOR BASIC TELETEX

UNIT	DEFINITION
Document User Information (DUI)	Smallest quantity of information preserved as one unit by the procedure.
Page (P)	Unit of information presented on one physical page.
Document (D)	A Sequence of one or more pages delivered as a single unit in the original page sequence.

(b) DIALOGUE ELEMENTS

UNIT	DEFINITION
Recovery Unit (RU)	A unit, delimited by a recovery mark inserted by the sender in the text stream, to provide a reference point for resuming transmission after a transmission interruption.
Commitment Unit (CU)	Element inviting the sink to take over responsibility for the transmitted information.
Delivery Unit (DU)	One or more commitment units that are to be considered as a single entity for the purposes of synchronization.
Interaction Unit (IU)	Element indicating the change of source and sink of data.

The dialogue elements are related in accordance with the following hierarchy:-

Document Level (RU - lowest level in the hierarchy
(CU
(DU

Session Level (IU - highest level in the hierarchy (started with CSS/CSCC ended with CSE/CSA/CSCC)

Initiation/termination of any unit in this hierarchy also initiates/terminates all units at lower levels of the hierarchy.

TABLE 1 (Continued)

(c) DOCUMENT LAYER COMMANDS AND RESPONSES IN TERMS OF THE ELEMENTARY FUNCTIONS

COMMAND/RESPONSE	USER STRUCTURE MEANING FOR BASIC TELETEX	DIALOGUE
CDS	Start of D, P	Start of DU (CU, RU)
CDE	End of D, P	End of DU (CU, RU)
CDR	Suspend D Discard current P	End of DU (CU, RU)
CDPB	End of P Start of P	End of CU (RU) Start of CU (RU)
CDUI	Start of DUI	
CDC	Continue D Start of P	Start of DU (CU, RU)
CDD	Discard D	End of DU (CU, RU)
RDEP	-	Ack (CU, RU)
RDPBP	-	Ack CU (RU)
RDPDN	-	Nack CU (RU)
RDRP	-	Ack DU (CU, RU)
RDGR	-	End of CU (CU, RU)
CDRP	-	Start of RU
RDRPN	-	Nack of RU
CDRPR	-	Start of RU
RDRPR	-	Start of RU

TABLE 2 CODING OF COMMAND AND RESPONSE HEADERS

(a) DOCUMENT LEVEL

Commands and responses are assigned in such a way that the bits may be interpreted as follows:-

Bit 1	1 = Command		0 = Response
2	1 = positive		0 = negative (for responses)
3	1 = initiate		0 = stop (for most commands)
4,5,6	1 1 1, 1 1 0, 1 0 1		Document/delivery unit
	1 0 0		(Reserved)
	0 1 1		Page (commitment unit)
	0 1 0		(Reserved)
	0 0 1		(Reserved for recovery unit)
	0 0 0		Text
7,8	set to zero and reserved for future extension		

The table shows, as an example, the bit allocation for document level commands.

COMMAND	BIT NUMBER 8 7 6 5 4 3 2 1
CDS	0 0 1 0 1 1 0 1
CDC	0 0 0 1 1 1 0 1
CDE	0 0 1 0 1 0 0 1
CDR	0 0 0 1 1 0 0 1
CDD	0 0 1 1 1 0 0 1
CDPB	0 0 1 1 0 0 0 1
CDCL	0 0 1 1 1 1 0 1
CDUI	0 0 0 0 0 0 0 1
CDRP	0 0 1 0 0 0 0 1
CDRPR	0 0 1 0 0 1 0 1
Reserved for private use	1 1 1 1 . . . .

(b) SESSION LEVEL

Commands and responses are assigned in such a way that the bits may be interpreted as follows:-

Bit 1	1 = command		0 = Response
2	1 = positive		0 = negative (response)
3	1 = initiate		0 = stop (for most commands)
4,5	1 1 Session		
	1 0 Session		
	0 1 Interaction		
	0 0 Session user		
6,7,8	set to zero and reserved for extension.		

TABLE 3 SESSION COMMANDS AND RESPONSES

Note: M = Mandatory for the basic Teletex service
NM= Non-mandatory

SESSION COMMAND/RESPONSE IDENTIFIER	PARAMETER GROUP DESCRIPTION (PGI)	M OR NM	PARAMETER DESCRIPTION (PI)	M OR NM
	SESSION INITIATION			
CSS Commands Session Start (CSS)	Session Reference	M	Terminal identifier of the calling terminal	M
			Date and time	M
			Additional session ref No.	NM
	Non-basic Session Capabilities	NM	Window size	NM
			Miscellaneous session capabilities	NM
	Non-basic terminal Capabilities	NM	Control character sets	NM
			Page formats	NM
			Miscellaneous terminal capab	NM
	Private Use	NM		
			Service Identifier	M
Response Session Start Positive (RSSP)	PARAMETERS AS FOR CSS			
Response Session Start Negative (RSSN)	For further study			
	Private Use	NM		
	TRANSFER PHASE			
Command Session User Information (CSUI)				

TABLE 3 (Continued)

SESSION COMMAND/RESPONSE IDENTIFIER	PARAMETER GROUP DESCRIPTION (PGI)	M OR NM	PARAMETER DESCRIPTION (PI)	M OR NM
Response Session User Information (RSUI)			Request session functions	NM
Command Session Change Control (CSCC)				
Response Session Change Control Positive (RSCCP)				
TERMINATION				
Command Session End (CSE)			Session termination	NM
Response Session End Positive (RSEP)				
Command Session Abort (CSA)			Session termination	NM
Response Session Abort Positive (RSAP)				

TABLE 4 DOCUMENT COMMANDS AND RESPONSES

Note: M = Mandatory for the basic Teletex service
NM = Non-mandatory

DOCUMENT COMMAND/RESPONSE IDENTIFIER	PARAMETER GROUP DESCRIPTION (PGI)	M OR NM	PARAMETER DESCRIPTION (PI)	M OR NM
INITIATION				
Command Document Capability List (CDCL)			Storage capacity negotiation	NM
			Graphic character sets	NM
	Non-basic Terminal Capabilities	NM	Control character sets	NM
			Page formats	NM
			Miscellaneous terminal capab	NM
	Private Use	NM		
Response Document Capability List Positive (RDCLP)			Parameters as for CDCL	
			Acceptance of CDCL parameters	NM
Command Document Start (CDS)			Document reference number	M
			Service interworking ident	NM
			Document type identifier	NM
	Non-basic Terminal Capabilities	NM	Control character sets	NM
			Page formats	NM
			Miscellaneous Terminal capab	NM
	Private Use	NM		

TABLE 4 (Continued)

DOCUMENT COMMAND/RESPONSE IDENTIFIER	PARAMETER GROUP DESCRIPTION (PGI)	M OR NM	PARAMETER DESCRIPTION (PI)	M OR NM
Command Document Continue (CDC)	Document Linking	M	Document reference number	M
			Checkpoint reference number	M
	only required if linking in a new session		(Called ID (	M
			(Calling ID (	M
			(Date and time (	M
			(Additional session ref No	NM
			Service interworking ident NM	NM
			Document type identifier	NM
			Document reference No. (current session)	M
			Other parameters of CDS	NM
TRANSFER				
Command Document User Information (CDUI)				
Command Document Page Boundary (CDPB)			Checkpoint reference number	M
Response Document Page Boundary Positive (RDPBP)			Checkpoint reference number	M
			Storage overflow warning	M
TERMINATION				
Command Document End (CDE)			Checkpoint reference number	M

TABLE 4 (Continued)

DOCUMENT COMMAND/RESPONSE IDENTIFIER	PARAMETER GROUP DESCRIPTION (PGI)	M OR NM	PARAMETER DESCRIPTION (PI)	M OR NM
Response Document End Positive (RDEP)			Checkpoint reference number	M
Command Document End Discard (CDD)			Reason (Document)	NM
Response Document Discard Positive (RDDP)				
Command Document Re-synchronize (CDR)			Reason (Document)	NM
Response Document Re-synchronize Positive (RDRP)				
ERROR RECOVERY				
Response Document Page Boundary Negative (RDPBN)			Reason (Document)	M
Response Document General Reject (RDGR)			Reflect parameter values	M
As mentioned in Paragraph 3.5.2 and 3.5.3 commands and responses from the transfer and termination have meaning in terms of error recovery.				

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

THE TELETEX SERVICE IN THE FEDERAL REPUBLIC OF GERMANY

Dipl.-Ing. Rolf Rüggeberg

Fernmeldetechnisches Zentralamt
Darmstadt

With the handing over of the Telecommunication Report by the KtK (Commission for Development of the Technical Communication System) to the Federal Government in 1976 the Deutsche Bundespost started its definite planning of new telecommunication services such as Telefax, Teletex and Bildschirmtext (CCITT term: Videotex).

Under the special aspect of office communications the main features of the Teletex service are presented. Its influence on existing services as Telex and Telefax, possible complementation or substitution of and interworking with other textcommunication services are dealt with briefly.

Teletex will offer new chances for the qualitative improvement of office organization. Including aspects of human engineering, possible organizational effects are considered.

Finally the decisions about the introduction of the Teletex service and the plans of foreign Administrations are presented.

1. RECOMMENDATIONS BY THE "COMMISSION FOR THE DEVELOPMENT OF THE TECHNICAL COMMUNICATION SYSTEM" (KTK)

The KtK was set up in 1974 by the Federal Minister of Posts and Telecommunications and had the task to study the possibilities of developing and expanding the technical communication system in the broadest sense and to work out concrete proposals.

Two years later the KtK handed over the results of its study to the Federal Government in its so-called Telecommunication Report and thus fulfilled its mission.

The results of the study were processed in such a way that they allowed conclusions to be drawn and, if possible, recommendations to be given for each telecommunication service. The entire field of telecommunications was subdivided into four categories:

1. Existing telecommunication services
2. New telecommunication services in existing networks
3. Telecommunication services in broadband distribution networks

4. Telecommunication services in broadband switching networks.

The statements and recommendations for the new textcommunication services Telefax, Teletex and Bildschirmtext are listed under category 2 "New telecommunication services in existing networks". For the Telefax and Teletex services, the KtK recommends to introduce these services, whereas it confines itself to the following statement about the Bildschirmtext service (1976):

> "The combined use of the public switched telephone network with home TV receiver enables new forms of telecommunications in existing networks. The information may consist of texts and still pictures. The subscriber will have the possibility, by means of an individual call, to occasion the reception of the information he has chosen from a variety of programmes offered. The technical prerequisites and the economic forms of utilization should be carefully observed and promoted."

Almost four and a half years after the publication of the Telecommunication Report the recommendations given by the KtK have been put into practice to a high degree:

- The Telefax service was introduced on 1 January 1979 using equipment of CCITT group 2 (3-minute transmission).
- The features of the Teletex service were for the first time presented at the Hanover Fair in 1980. In 1981 the service will be introduced within the area of the Deutsche Bundespost, Sweden and Canada will follow in the same year.
- Since June 1980 a large-scale field trial has been going on in Berlin and Düsseldorf with about 5000 subscribers of the Bildschirmtext service. The start of this service is scheduled for 1982.

In the following our considerations will concentrate on the development of the Teletex service under the special aspect of office communications. In comparing the Teletex service with the existing textcommunication services "Telefax" and "Telex" and looking at interworking possibilities with other services, we touch upon a few particularities of the other textcommunication services.

2. INTERNATIONAL STANDARDIZATION OF THE TELETEX SERVICE IN THE CCITT

From the beginning, the Deutsche Bundespost has attached great importance to the fact that the new Teletex service will be an internationally compatible textcommunication service.

A new Study Question was adopted by the CCITT (Comité Consultative International Télégraphique et Téléphonique), and since 1977 Study Groups I (Recommendation for the Teletex service) and VIII (Recommendation for the Teletex terminal) have been engaged in laying down the conditions for the international Teletex service.

The work in these bodies was concluded in June 1980, and the Draft Recommendations were adopted by the Plenary Assembly at its last meeting in November 1980 in Geneva.

For the Teletex sector the following Recommendations have been worked out:

Rec. F.200	Teletex Service
Rec. S.60	Terminal Equipment for Use in the Teletex Service
Rec. S.61	Character Repertoire and Coded Character Sets for the International Teletex Service
Rec. S.62	Control Procedures for the Teletex Service
Rec. S.70	Network Independent Basic Transport Service for Teletex

3. FEATURES OF THE TELETEX SERVICE (Extract from CCITT Rec. F.200)

3.1 BASIC DEFINITIONS

- The Teletex service allows the exchange of information between the subscribers in an automatic memory-to-memory system.

- Basically it should be possible that the Teletex terminal is operated like a normal office typewriter.

- Operation of the keyboard and text production shall not be subject to any restrictions and be possible in the same way as it is usual with office machines.

3.2 TRANSMISSION RATE

Transmission rate of the international Teletex service was fixed on 2400 bit/s.

3.3 RECEIVING MEMORY

The Teletex terminal must be equipped with a receiving memory in order to guarantee uninterrupted local operation and to bridge the different transmission rates during the transmission and printing processes.

3.4 CHARACTER REPERTOIRE

To guarantee worldwide compatibility, a basic character repertoire was defined, which has to be "understood" by each Teletex terminal. This basic character repertoire is the entire Latin graphic character repertoire. In a first approximation it is the sum of all characters which can be generated by office typewriters in countries using Latin graphic characters.

With respect to the basic character repertoire, the Teletex terminal must meet the following requirements:

In the phase of text production only those characters are generated, which are available on the keyboard used (= sub-repertoire of basic character repertoire).

For reception it is necessary that <u>all</u> characters of the basic character repertoire are represented.

3.5 CALL IDENTIFICATION LINE

Analogous to the Telex service, it is required for the Teletex service that, after the call setup and before text transmission, the called subscriber is identified in order to ensure that the connection with the wanted subscriber exists and wrong connections are impossible.

The information required for this purpose is called reference information and represented in one line, the so-called call-identification line (Fig. 1).

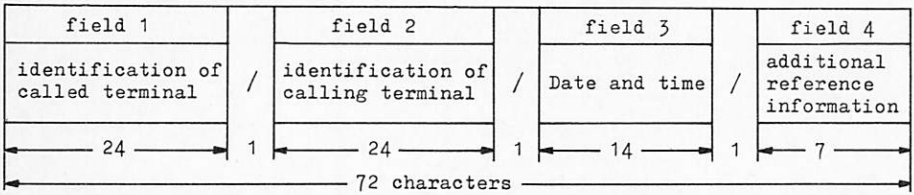


Fig. 1 Call-identification line

3.6 IDENTIFICATION OF THE TERMINAL

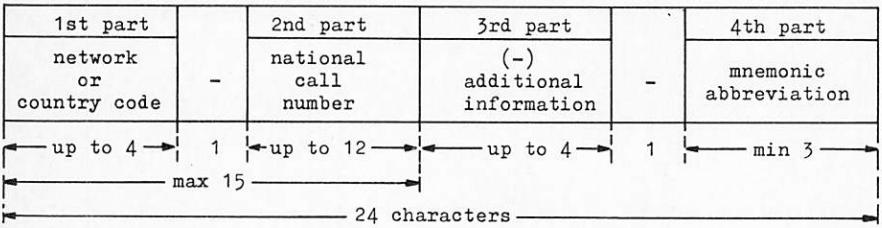


Fig. 2 Identification of terminal

3.7 FORMAT

The printable area on a page is determined by the fact that in the international service both the A 4 format (210 x 297 mm) and the North American format (216 x 280 mm) are permitted. For the area being common to both formats, the size of the printable area is defined in such a way that the maximum possible number of lines and the maximum possible number of characters per line were specified at a character pitch of 2.54 mm (= basis).

orientation of page		vertical (A 4)	horizontal (A 4 L)
number of lines	line feed 1-line 4.23 mm 1 1/2-line 6.35 mm 2-line 8.47 mm	56 37 28	40 26 20
number of characters per line	character spacing 2.54 mm	77	92

Fig. 3 Printable area of a page

4. TELETEX AND TELEX

4.1 INTERWORKING BETWEEN THE SERVICES

The capability of interworking with the Telex service was one of the basic requirements the Teletex service had to meet. This meant in detail:

1. Interworking between a Telex and a Teletex terminal must take place automatically without any assistance of operators.

2. Interworking shall be performed via a conversion unit in the network.

3. In international traffic use shall be made of 50-baud telex circuits, i.e. the conversions are always effected on the Teletex side. (The use of international Teletex connections (2.4 kbit/s) shall later on be possible after bilateral agreement).

4. In a Telex/Teletex connection the operating mode on the Telex side shall largely correspond to that of conventional Telex operation.

5. The grade of service on the Teletex side must not be deteriorated by the interworking with Telex.

Fig. 4 shows the basic configuration for Telex/Teletex interworking, the international link being a 50-baud telex circuit. The operational procedures have already largely been defined taking account of the above requirements.

According to them the Teletex terminal works in a so-called telex mode of operation where the Teletex terminal is allowed to use only characters of the International Telegraph Alphabet No 2 and to print a maximum of 69 characters on one line (= telex format). All other conversions such as

- code conversion (7-bit code → 5-bit code)
- rate conversion (2.4 kbit/s → 50 bit/s)
- procedure conversion (Teletex procedure in Telex procedure)

are performed by the conversion unit in the network. The same applies to the opposite direction (Telex - Teletex).

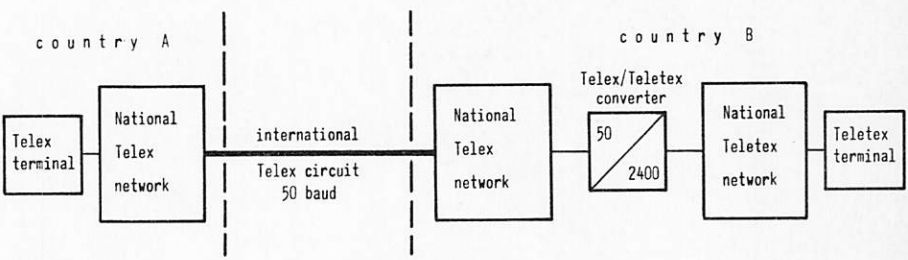


Fig. 4 Basic configuration for interworking with Telex

The rate conversion necessitates that the conversion unit has to be equipped with a store for buffering the messages. By that it is guaranteed that the Teletex subscriber line is not occupied longer than in the case of a normal Teletex connection. This means that the grade of service criteria described in Section 3.3 are not affected by the interworking with Telex.

4.2 INFLUENCE OF TELETEX ON TELEX

An exact analysis of the future effects of the Teletex service on the Telex service is not possible yet at the time being, as some factors and parameters are still unknown, e.g.

- the cost of the Teletex terminal,
- the development of the costs of the Telex terminal after introduction of the Teletex service.

On the assumption that the number of Teletex terminals will be about 40 000 in 1985 and increase to about 130 000 in 1990, it is expected that

- the Telex service will not be affected noticeably in the first years following the introduction of the Teletex service,
- from 1985 the number of Telex terminals will decrease by 10 000 annually (Fig. 5).

Although the number of Telex terminals will diminish, the Telex service itself and by that Telex traffic will continue to be of great importance, particularly at the international level.

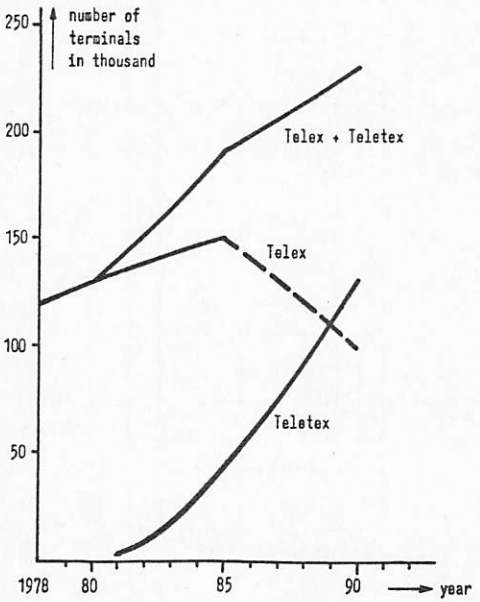


Fig. 5 Estimate effect of Teletex on Telex

5. TELETEX AND FACSIMILE

5.1 INFLUENCE ON THE TELEFAX SERVICE

The Telefax service differs considerably from the Telex and also from the Teletex services. Its main characteristic is that it allows the transmission of any type of written documents. Texts already existing do not have to be written again.

The system on which the Telefax service of the Deutsche Bundespost is based (equipment of CCITT Group 2, analogue transmission in the telephone network) does not permit any mixed mode of operation with the future Teletex service, i.e. the joint use of character- and facsimile-coded information is not possible. Consequently, the Telefax service will exist parallel to the Teletex service, especially since the increase in the demand for this service is expected to continue. The Telefax service covers special needs in the field of office communications and is used for applications like the transmission of existing texts and attachments, manuscripts, drawings, etc. Both services, Telefax and Teletex, will be of growing importance for the office of the future.

5.2 POSSIBILITIES OF FUTURE FACSIMILE TECHNIQUE

Interworking between facsimile and Teletex equipment will only then be possible when the standardization of group-4 equipment (digital transmission, use of data networks) has been completed.

The problem involved in interworking between the two services is whether a Teletex terminal can, in addition to character-coded, receive and represent also facsimile-coded information and thus is suitable for a mixed mode of operation.

Here a wide field of applications opens up, which are at present being investigated and probed with regard to their effects and consequences. There are not precise conceptions yet in which way character- and facsimile-coded information can be used in "the office of the future" and how the demand and cost situations will develop.

But it can already be foreseen that

- for the phase of text production a mixed mode of operation will not be necessary (character- and facsimile-coded information will be generated separately)

- for the phase of transmission and thus also for the phase of reception there will be a demand for mixed modes of operation.

As an example it may be mentioned that fairly often there is the need

- to insert texts into drawings or to inscribe them,
- to write texts on forms
- to enclose texts already existing as annexes.

In the CCITT a possible future mixed operating mode was taken into account in such a way that, as far as possible, common transmission procedures were defined for character- and facsimile-coded transmissions.

6. TELETEX AND BILDSCHIRMTEXT

Type and extent of interworking with the Bildschirmtext service (provisional CCITT term: Videotex) is yet to be determined. A new Study Question was formulated at the Final Meeting of Study Group I in Montreal (June 1980), and it can be expected that the basis for interworking between the two services will be provided during the next Study period.

One thing is certain - interworking between the Teletex and Bildschirmtext services will be confined to that part which is common to both services: the transmission of texts.

It is to be expected that the future Teletex subscriber will be supplier (and also receiver) of information in the Bildschirmtext service and will have access to the Bildschirmtext centres. The prerequisites that the character repertoire in the Bildschirmtext service corresponds to that of the Teletex service and that an unambiguous assignment of the character repertoires exists have already been created.

7. INVESTIGATIONS OF THE EFFECTS ON OFFICE ORGANIZATION

7.1 NEGATIVE EXPERIENCE IN OFFICE RATIONALIZATION

It is often claimed that office work has considerable rationalization reserves. While, for instance, during the last 100 years productivity on the manufacturing sector has increased by 1400 %, the increase on the office sector over the same period amounted to about 100 %. Thus, it was an obvious conclusion that the application of new office techniques would lead to an abrupt increase in office work productivity. The experience gained in past years with new office techniques have not always been positive. As an example, we may mention the centralization of the typing services. The work at the typist work place is relatively easy to quantify. By forming typist pools the productivity of a typist work place has indeed been raised noticeably. But an overall judgement has to take account of other effects as well. For instance, by separating the typist service from those giving the writing orders and by taking it out of the organizational context, so-called "pass-on" effects arise, where activities are shifted and not included in the economic effiency considerations:

- office activities are switched from the typist to the senior clerk
- distance to be covered to the central typist pool,
- non-typists are entrusted with typing work
- urgent documents are typed by the senior clerks themselves.

The effects have a negative influence on the total economic efficiency and occur where typist services have been centralized.

7.2 NEEDS OF OFFICE PRACTICE

The needs of practical office work are less oriented towards higher quantitative performance than towards structural and qualitative improvements. Complaints are generally expressed about

- the poor transparency of administrative activities
- low and inefficient handling of work
- long official channels
- bureaucratic and formalistic regulations
- little flexibility in the case of outside changes

- poor coordinating capabilities
- high administrative expenses.

Rationalization considerations should therefore not only aim at one-sided productivity increases, but they must also include structural and qualitative improvements. It must be attempted to set a differentiated efficiency approach which includes the qualitative aspects in the economic appraisal against a superficial economy thinking solely based on economic efficiency data.

7.3 TELETEX FIELD TRIALS

7.3.1 GENERAL

At present several large-scale field trials are being carried out by the industry (supported by the Federal Minister for Research and Technology) in which the possibilities of improving office organization including qualitative and structural aspects are investigated. The following projects are under way:

Field trial textcommunication Allianz
User: Allianz Insurance Comp.
Manufacturers: AEG-Telefunken, T&N, Olympia AG

TEKOM field trial (Siemens AG).

The objective of these trials is, in addition to testing the equipment under operational conditions, to obtain above all the following information on the following points:

- acceptance of the new service
- economic benefit including structural and qualitative aspects,
- repercussions on existing organization
- effects on the individual and chances for the utilization of a wide range of communications offered.

The socio-scientific side of these projects is handled by the Hanover University and the Bundeswehr Academy in Munich.

7.3.2 FIRST RESULTS

The procedure for the Teletex field trials was divided into three phases:

Phase I — Investigation of the existing organization before the implementation of the new technique. After that installation of Teletex terminals and instruction of the operators about the local functions

Phase II — Examination of the local functions by the operators. After that starting with the communication

Phase III — Examination of the technique used for the communication and the operation. Investigation of the impact on office organization.

Phase I and II were finished at the end of 1980, phase III has begun in January 1981.

The results of the investigation carried out in phase I and II allow some basic statements about future expectations in conjunction with the introduction of the Teletex service:

- The relation between internal and external communication of the enterprises concerned is about 6 : 1. Consequently big organizations will use the Teletex service first to handle the internal communication.

- There is a demand for an increasing usage of technical communication facilities such as Telex and Telefax. This existing demand will facilitate the application of the Teletex service in future.

- A dezentralized location of the Teletex terminals will offer good chances to make office organization more effective and improve the working situation of the typing staff considerably.

8. INTRODUCTION OF TELETEX SERVICE

After the successful demonstration of the Teletex features at the Hanover Fair (April 1980), in Montreal (June 1980) and after the conclusion of international standardization in the CCITT, the following schedule has been drawn up by the Deutsche Bundespost:

- until end of February 81 continuation of field trial on the basis of the Hanover Fair technique

- in March 81 starting of the Teletex service with subscribers on the basis of the Hanover Fair technique

- autumn 81 implementation of the CCITT technique (network and terminals).

At the present time the following European Administrations have announced that they intend to introduce the Teletex service: Belgium, Denmark, France, Italy, Netherlands, Norway, Sweden and United Kingdom.

Moreover, a number of countries outside Europe have expressed great interest in the Teletex service (e. g. Canada, USA, Japan, Australia).

REFERENCES:

KtK-Report (Telekommunikationsreport der "Kommission für den Ausbau des technischen Kommunikationssystems", 1976)

CCITT Draft Recommendation F.200 "Teletex Service" of Study Group I

Picot/Reichwald
Teletex - Chancen für eine qualitative Verbesserung der Büroorganisation
(Telefonbau und Normalzeit, 1980)

Picot/Reichwald
Untersuchung der Auswirkungen neuer Kommunikationstechnologien im Büro
auf Organisationsstruktur und Arbeitsinhalte
Zwischenbericht, Dezember 1980

Rüggeberg
Teletex - ein neuer internationaler Textkommunikationsdienst
(Der Fernmelde=Ingenieur, 33. Jahrg., Heft 11, November 1979).

NAMING/ADDRESSING ISSUES

Chairman - P. Schicker

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

ADDRESSING AND DIRECTORY SYSTEMS FOR LARGE COMPUTER MAIL SYSTEMS

Jose J. Garcia-Luna and Franklin F. Kuo[1]
Dept. of Electrical Engineering
University of Hawaii at Manoa
Honolulu, HI 96822

In this paper, we present a general description of the addressing schemes necessary to provide efficient identification and delivery services in large, distributed computer mail systems and interconnected systems. The structure of the required directory system is described in detail under the assumption of the existence of a protocol to communicate the delivery processes. A naming standard and an addressing standard are introduced, and for the purposes of completeness the basic structure of the system components of our architecture is introduced at the beginning of this work.

I. INTRODUCTION

The use of computer networks for human communication is economically and technically feasible today. Almost every computer networking company is now offering or has plans to offer electronic message systems, and the message traffic today from commercial sources is of the order of several million messages annually [8], [11], [12]. Because of the many benefits computer mail offers [6], [8], the number of computer mail users and systems is expected to increase significantly in the future, and soon many more computer mail systems with thousands of users will come into being. Unless adequate addressing, delivery and message processing standards for computer mail are promulgated, those computer mail communities will be isolated from each other and even users within the same systems will face many difficulties in trying to communicate with one another. Unfortunately, it appears that the techniques currently used to implement computer mail systems are suitable neither for larger communities nor for the interconnection of heterogeneous systems, and the problems that arise in developing addressing, delivery and message processing standards under such circumstances are yet to be solved.

In this paper we address the design of addressing and delivery schemes necessary to provide reliable message delivery and on-line identification services in large computer mail systems and interconnected systems. First, the main concepts of distribution of functions for computer mail systems are presented. Then we outline the basic addressing and delivery techniques of our model without detailing the communication protocols involved. Instead, we focus on the procedures by which messages are delivered and addresses are obtained and introduce a naming and an addressing standard. Finally, the directory system that maintains all identification information is described and its role in the delivery process is further clarified.

II. SYSTEM FUNCTIONS AND COMPONENTS

Here we will focus on large computer mail systems and unions of computer mail systems and describe the components of an architectural model which specifies the user functions of such systems. For the specification of the proposed architecture we rely on the assumption that an underlying "implementation

level" provides the functions taken for granted by the architecture, such an implementation level is made up of those processes which provide the transport services that enable the various components of the computer mail system to communicate with each other in order to provide the services expected by the users of the system.

Our architecture is defined at the application level and is formed by a set of independent functional components, each of which supports a specific set of functions to be provided. The functional components to be defined constitute processes which should not be directly related with underlying implementation considerations such as host computers, intelligent terminals, or communication networks.

We define three functional components in a computer mail system: mailboxes, mailers, and gateway mailers.

MAILBOX (MBX) is the process dedicated to the creation and retrieval functions of the system; this is also the interface between user and delivery service, that is, all user's messages are created by the interaction between the user and his mailbox. A mailbox is formed by three elements, namely:

1. The user interface
2. The mailbox database
3. The mailer interface

A personal directory with system addresses of recipients is maintained in the mailbox database together with any other information useful for the user, such as previous incoming and outgoing messages. The user interface is the process by which the user creates and retrieves messages. The mailer interface is the process by which the mailbox communicates with its local mailer.

MAILER (MLR) is the process that manages the delivery of messages to and from mailboxes and communicates when necessary with other mailers or the gateway mailer of the mailing network. Each mailer serves a set of one or more mailboxes, called local mailboxes, and each mailbox communicates with one and only one mailer, called the local mailer. A mailer is formed by four elements:

1. The mailbox interface which communicates with the local mailboxes
2. The network interface which communicates with other mailers
3. The mailing protocol handler which is in charge of the set of rules for the delivery of messages to and from the local mailboxes
4. The mailer's directory database which contains addressing information as well as control information about the negotiations handled by the mailer

GATEWAY MAILER (GMR) is the process that manages the delivery of messages between a mailing network and other mailing networks connected to it. Each gateway mailer serves a unique set of local mailers and each local mailer communicates with one and only one gateway mailer, called the local gateway mailer. This component performs all the necessary transformations of format, addressing and data representation to communicate the local mailing network with others. The structure of a gateway is basically the same than that of the mailer, namely it consists of:

1. The gateway directory database
2. The mailer interface
3. The system interface
4. The mailing protocol handler

The primary role of a gateway mailer at the intrasystem level is to serve as an intermediate hop between source and destination mailers to alleviate the identification tasks of each mailer in a network; at the intersystem level, its role is to support and enforce the independence and cooperation of intersystem mailing processes.

MAILING NETWORK is the union of logically connected mailers (implying both reachability and addressability). The topology of a mailing network (MNT) is totally independent of the underlying communication network's topology. The delivery of a message within a mailing network is called intranet delivery. When more than one mailing network is involved the process is called internet delivery. Each MNT has associated with it a gateway mailer (Fig. 1).

COMPUTER MAIL SYSTEM is the union of logically connected mailing networks offered by the same server, and the corresponding mailboxes (Fig. 2). Such a system might be formed by the interconnection of several packet-switching networks such as Telenet, TYMNET, DATAPAC and TRANSPAC, or by the interconnection of several kinds of services such as packet switched data, Teletex, Prestel and Facsimile.

III. ADDRESSING AND DELIVERY

Background

Users compose messages to be delivered to other users, groups of users, organizations, processes, etc., that we call recipients. Even though messages are meant to be delivered directly to recipients, the actual delivery of the messages is not made to the recipients themselves but to some logical entities or processes which we call mailboxes. In our architecture the population of mailboxes in the system is viewed as a set of named objects and each recipient is associated with a mailbox for message delivery.

For messages to be delivered, the recipients' system mailbox addresses must be first obtained; this implies the solution of three problems: how to specify a system mailbox address, how to determine a recipient's system mailbox address, and how and where to maintain the system mailbox addresses of all recipients in the system. In small centralized systems with a few hundred users accessing the same computing facility, the identification of an address is not difficult, e.g., system mailbox addresses can be directly associated with users' names. But the situation radically changes when large user communities are involved (e.g., tens of thousands of users), various systems are interconnected, and the system(s) spans various countries and organizations. In these cases we can expect to have:

1. Wide variations in formats, data models and amounts of information maintained in different databases.
2. Large amounts of identification information maintained in different databases.
3. Widely varying identification services offered.
4. Differences in addressing and delivery standards between systems.
5. Differences in privacy and security measures between systems, organizations and countries.

Previous approaches to address specification and address identification probably would not work for identification services in large, distributed computer mail systems. The approaches taken by the postal service and the telephone system are outlined in [7], and we will only point out various inconveniences of such approaches. The postal service does not provide any address identification service at all; the specification of postal addresses has been based on a

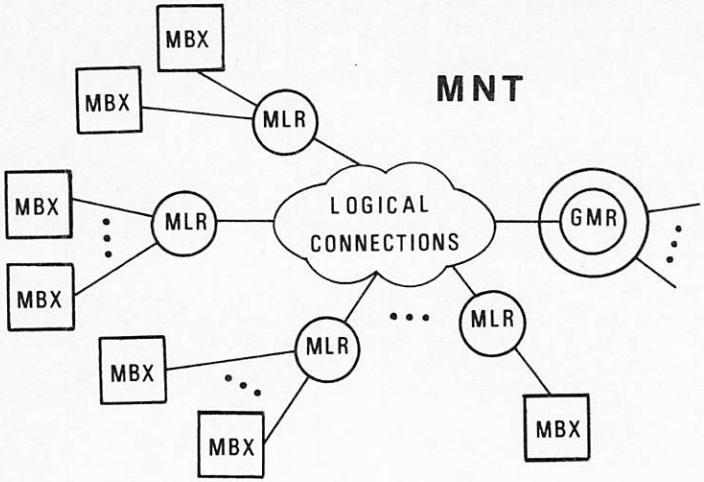


Figure 1 - MAILING NETWORK

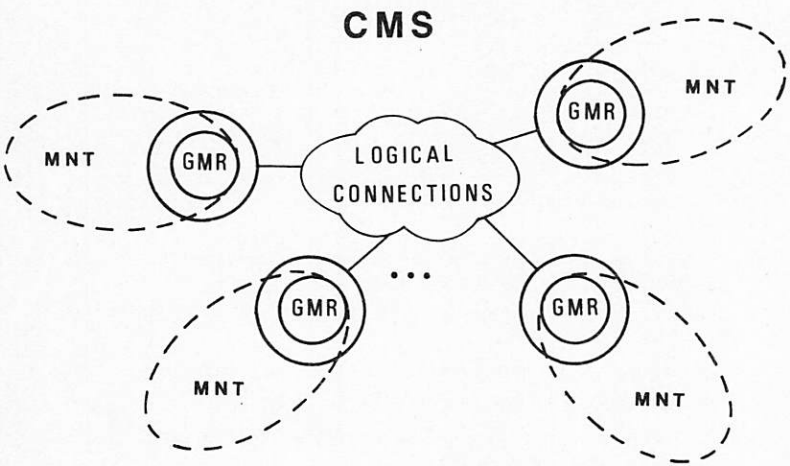


Figure 2 - COMPUTER MAIL SYSTEM

geographical partitioning of the user community (though some large companies have been given their own zip codes and the number of those cases is expected to increase when the 9-digit zip code format is adopted); and the delivery of postal mail heavily relies on intelligent human decisions. The telephone system supports limited identification services by providing listings of users by names and categories (services) and listings of telephone numbers by street addresses. But because the limited information that is maintained in telephone directories and because of their rigid logical structure, the system fails too often [7]. The approach of the telephone system to the address specification problem does not work well at the local level [7], where organizations have various extensions of the same telephone number and various users and/or devices share the same number.

To the best of our knowledge, only one computer network has come out with a practical approach to the problem of address identification--the ARPANET [2], [4], [5], with its Network Information Center (NIC). Addressing of mailboxes in ARPANET is done according to a two-level address format [4], [5], [7], of the form <mailbox name>@<host computer>. This approach is acceptable when the number of hosts in the network (which in ARPANET support both delivery and message processing functions) is small, e.g., less than a hundred. The NIC's identification system identifies users by their last names and NIC IDENT codes. The NIC also publishes the ARPANET directory annually. However, many of the ARPANET's users are not included in the identification database and the structure of the information is the same for all of the users in the network.

If tens of thousands of users exist in a federation of computer mail systems and various countries and companies are included, the structure of the identification information maintained in the systems must be flexible to allow for the description of users in different milieus. In computer mail systems the location of recipients can be independent of the location of their mailboxes; and with the advent of satellite communications, geographical boundaries do not represent a major factor in computer mail system operation. In the future, the availability of personal computers and local networks connected to larger computer networks will make two-level addressing schemes such as the ARPANET's difficult to apply. So, we propose a more flexible and general approach to identify and specify system mailbox addresses in large, distributed computer mail systems and interconnected systems.

Proposed Techniques

A normal sender of a large distributed computer mail system, or a federation of systems, would rarely know the system mailbox address of a recipient, but the sender might well be able to describe the recipient in terms of his name and/or his title in an organization, his organization, his geographical location and (perhaps) his computer mail system or network. In the proposed architecture, the senders of messages are allowed to describe the recipients in terms of user-oriented descriptions and not their system mailbox addresses; and the responsibility of the system is to map a user's description of a recipient into the correct system mailbox address in a form completely transparent to the users. Thus, users are freed from the task of knowing the addressing structure and procedures followed in the computer mail system(s) they access.

A common format must be used to communicate the users' descriptions of recipients so that the system components can understand the information provided by the users. We define a naming standard that we call the NOLS address [3] to communicate recipient descriptions in a user-oriented, machine-processable form. A NOLS address is a set of four flexible fields which describe the attributes of a set of one or more recipients. The four fields of a NOLS address are the following:

N-field -- Contains information about the recipient(s) of the message, such as his Name and/or title.
O-field -- Contains information about the Organizations(s), group(s) or system server(s)
L-field -- Contains information about the geographical Location of the organization(s) or group(s) referred in the O-field
S-field -- Contains information about the System which offers the computer mail services to the recipient(s).

By organization we mean any entity well-known to the server and which corresponds to a subset of the total community of users; thus, an organization can be: a company, a regional center, a regional computer mail server, a branch office, etc. We shall see in the next section that the contents allowed in each field of a NOLS address depends on the sophistication of the system's identification database.

We define the term NEXUS to be an end-to-end virtual connection between the sender's mailbox and the recipient's mailbox for the purposes of message delivery. Each NEXUS is identified via a pair of mailbox addresses, and a system mailbox address is specified in three layers, as is shown in Table 1.

When a sender enters a message through the user interface of his mailbox he also enters a NOLS address that contains what he knows about the recipient's name (and/or title), organization, geographical location and computer mail system. When the sender instructs his mailbox to deliver the message, the mailbox transfers the message to the local mailer, who is then in charge of its delivery. Figures 3 and 4 show the various steps followed to deliver a message (for both the internetwork and intranetwork cases). In order for a message to be delivered, its NOLS address has to be mapped into a set of one or more system mailbox addresses; thus, such a NOLS address serves as a query to the system for obtaining the correct system mailbox address(es). The local mailer carries out this mapping function in cooperation with other mailers and gateway mailers in the system by forwarding the NOLS address to the appropriate mailers or gateway mailers (Fig. 3) according to the information maintained in the directory system (to be described in the next section). When a remote mailer replies with a system mailbox address, the NEXUS between the sender's mailbox and the recipient's mailbox is created; and as is shown in Fig. 4, the NEXUS once established serves as a virtual information channel from the sender's mailbox to the recipient's mailbox for the delivery of subsequent messages.

Note that each field of a NOLS address corresponds to a partition of the total population of recipients. When the four fields are correctly and unambiguously specified, the ultimate partition uniquely corresponds to the intended recipient. Since users are allowed to ask for the delivery of messages regardless how little information they provide in the NOLS addresses, these addresses may in some cases be ambiguous, lack information in a certain field, or contain redundant information. In such cases several gateway mailers and/or mailers may have to be queried before the desired system mailbox address is obtained. In some cases, if the NOLS address is very poorly specified, the system may be unable to identify the recipient's system mailbox address at all.

In order for mailers and gateway mailers to communicate with each other and activate addressing and delivery functions, a standard set of commands must be defined. Table 2 presents the basic set of commands necessary for interprocess communication and Fig. 5 shows how these commands are exchanged between mailers and gateway mailers. Generally speaking, message delivery in our architecture is carried out in two phases: the NEXUS establishment phase, where the NEXUS between sender's mailbox and recipient's mailbox is created; and the message delivering phase, which consists of the actual delivery of sender's messages.

Table 1 - Addressing Formats

Format type	Components
Extended ARPANET address	<mailbox name>@<host name>@<network name>
Proposed 3-layer system mailbox address	[<internetwork layer><intranetwork layer><local layer>] Internetwork layer--Contains the name of the network specified according to system-wide formats. Intranetwork layer--Contains the name of the mailer in the mailing network specified in the internetwork layer. The mailer's name is specified according to formats particular to the network. Local layer--Contains the name of the mailbox in the mailer specified in the intranetwork layer. The name of the mailbox is specified according to formats particular to the mailer.
NEXUS address	([sender's system mailbox address] > [recipient's system mailbox address])

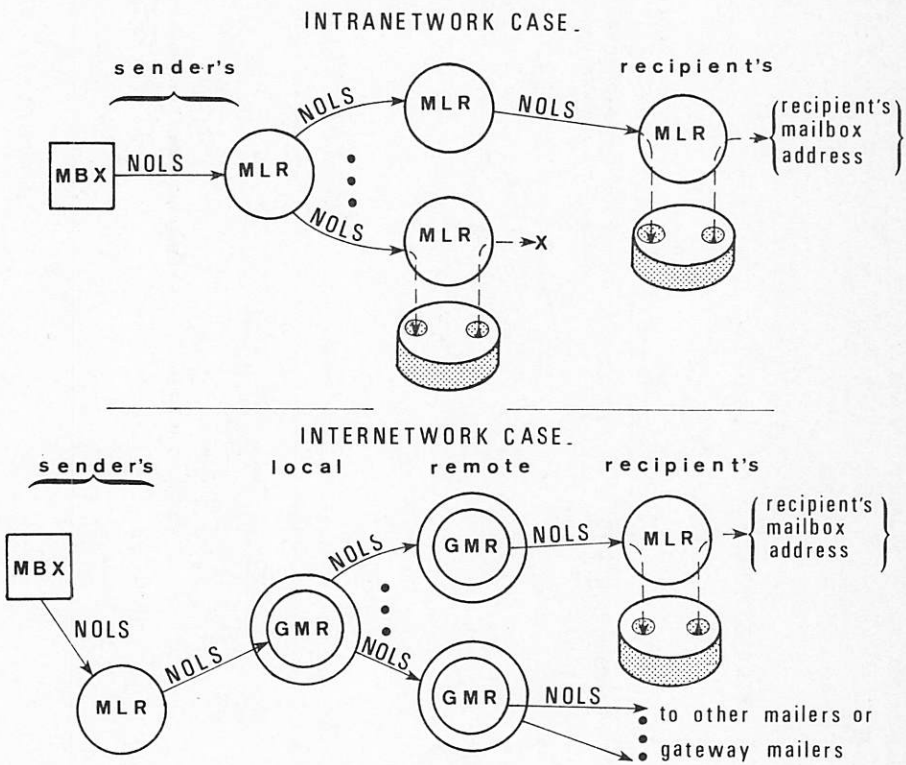


Figure 3 - FORWARDING OF NOLS ADDRESSES

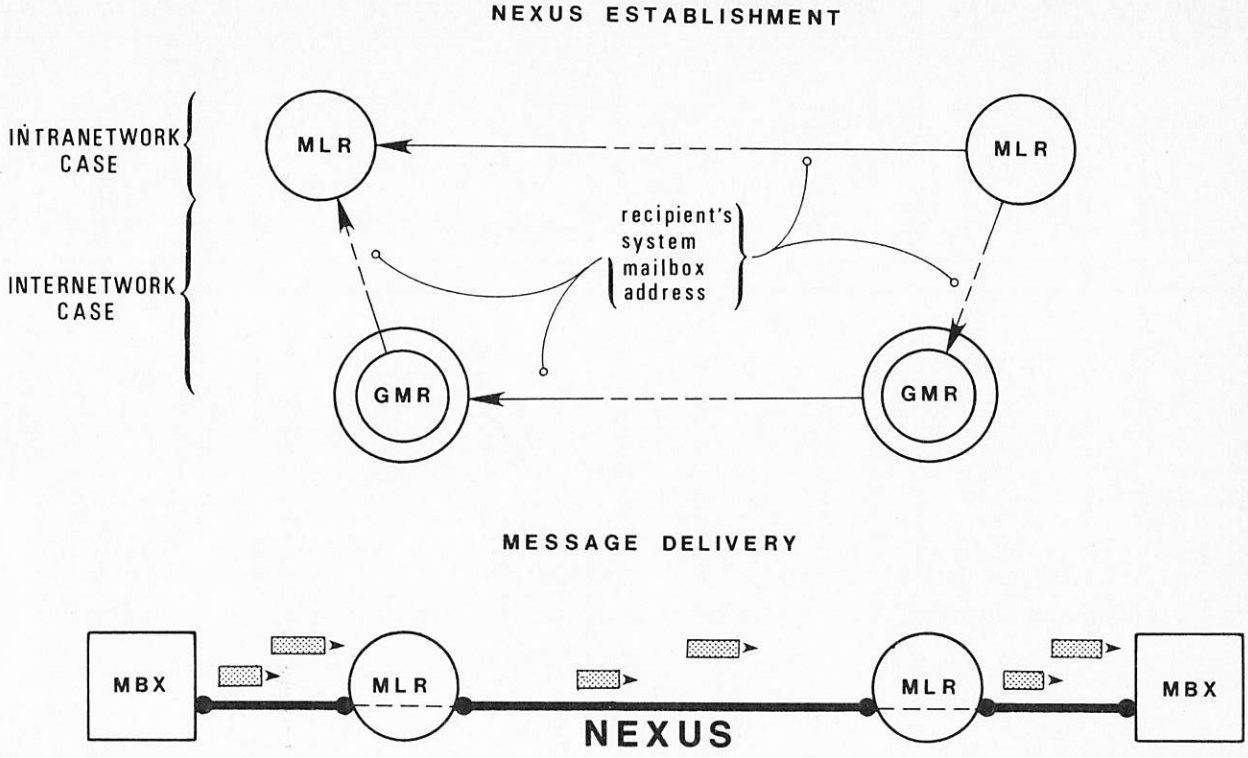


Figure 4 - MESSAGE DELIVERY

Table 2 -- BASIC SET OF COMMANDS FOR INTERPROCESS COMMUNICATION

DELIVER	-- Contains the user message, together with the NOLS and NEXUS address of both sender and recipient
IDENTIFY	-- Contains the user query, i.e., a NOLS address created by the sender. It also identifies the sender for management purposes.
ACK-ID	-- Contains the positive reply to an IDENTIFY command with the system mailbox address and NOLS address of recipient.
NACK-ID	-- Contains negative reply, possibly including a "list of similar names".
ACK-DEL	-- Contains positive acknowledgement to a successful delivery.
NACK-DEL	-- Contains negative acknowledgement to a DELIVER command and is possibly due to component failure, unauthorized sender, or non-existent mailbox address.
UPDATE	-- Calls for an update in the directory database and contains time-stamped information.

As shown in Fig. 5, the NOLS addresses entered by the senders are forwarded by means of IDENTIFY commands and messages are delivered by means of DELIVER commands. Note that the correct operation of the delivery protocol [3] is guaranteed by the provision of: a) forwarding capabilities that permit message delivery and address identification in case of change of address of recipients and/or process failure, and b) the proper identification (by means of NOLS addresses) of both the recipient and sender in every message delivery and identification query, as is shown in Fig. 5. Key points to note in the delivery process are:

1. A mailbox knows only about the existence of its local mailer.
2. A mailer knows only about the existence of its local mailboxes, other mailers and the gateway mailer in its mailing network.
3. A gateway mailer knows only about the existence of the mailers in its own mailing network and other gateway mailers in the total system.

IV. DIRECTORY SYSTEM

Background

A directory is a listing of identification information about recipients and subsets of the community of recipients. A directory system is formed by the union of directories in mailboxes, mailers and gateway mailers, plus the data and procedures necessary to manage the information in those directories. Thus, a directory system is a distributed identification database system.

The primary purposes of a directory system are to provide the users of the system with an integrated view of the identification information distributed in the computer mail system(s), and to allow for the reliable and efficient identification of recipients' system mailbox addresses.

There are four main issues in the design of a directory system:

1. How the directory system is organized.
2. What information is maintained in the directory system.
3. How the information is distributed and structured.
4. How the information is updated and processed.

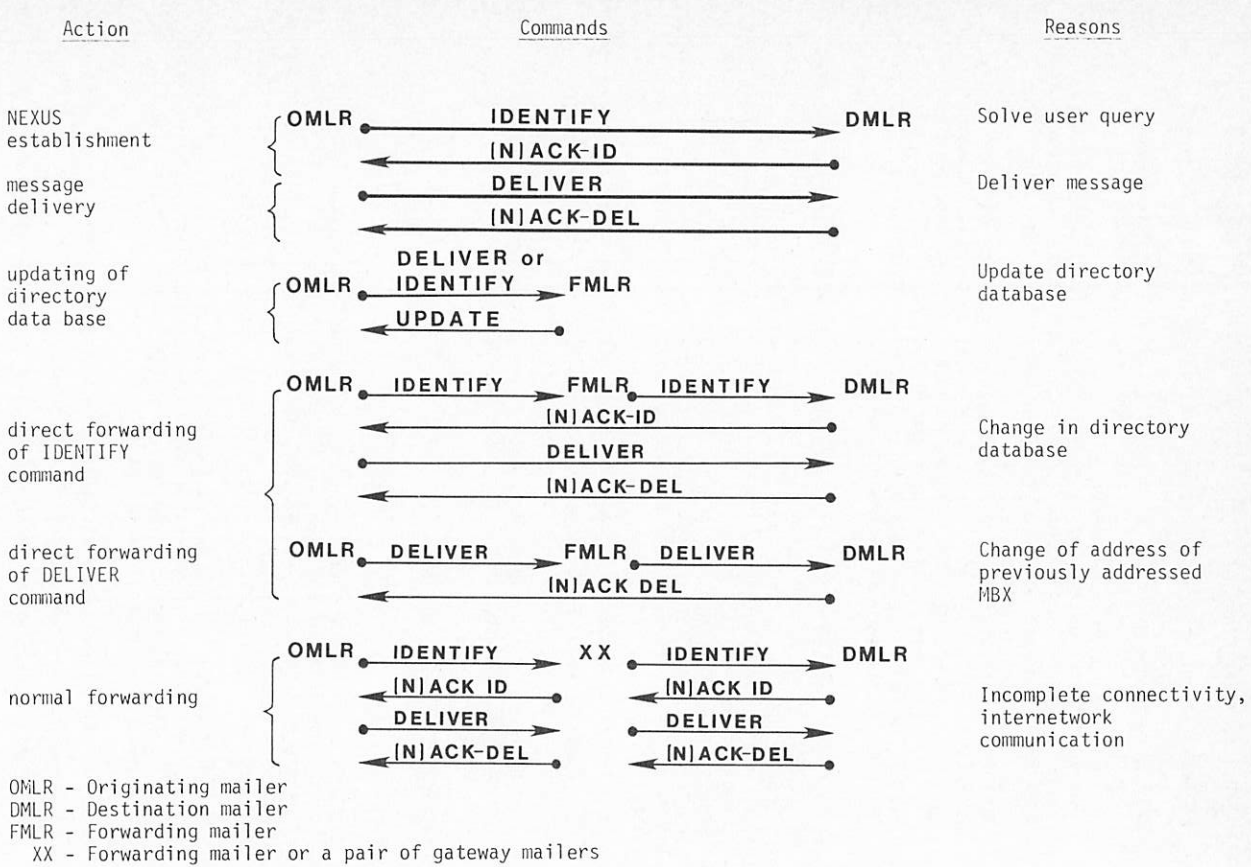


Figure 5 - INTERPROCESS COMMUNICATION

Directory System Organization

We propose a hierarchical organization of the directory system [3] to:

- ensure independence and enforce cooperation among system components;
- provide a user's view of the computer mail system with a simple structure;
- permit each organization in the computer mail system to maintain the identification information concerning its own members according to its own needs; and
- establish an interactive process between sender and system for the sender to obtain detailed addressing information of intended recipients.

With such an organization, all detailed information is maintained independently at each mailer, and the necessary data and procedures are maintained among mailers and gateway mailers to support their interaction. Users in the same mailing network are given an integrated network-wide view of the computer mail system, and all interaction between senders and system is based on that view, with both the distribution of information in the system and the different structures of the information maintained at different mailers being completely transparent to the users.

The hierarchical directory system we propose is organized in four layers, the highest layer being the "User layer", and each layer of the directory system serves as the means for the interaction between elements at the layer above. Figure 6 shows the organization of the directory system, whose layers are the following:

1. User Layer: This layer consists of personal directories maintained in the mailboxes. The management of each personal directory is carried out by the user interacting with his mailbox and is independent from the rest of the system.

2. Local Layer: This layer consists of the union of the local directory systems supported as part of each mailer (Fig. 6). The functions at the local layer of the directory system compose: the management and description of the logical and physical structure of the information about local recipients, the control of the physical access of this information, and the storage of the information. Each local directory system consists of: a local directory with complete identification information about all the local recipients, a forwarding directory with the mailbox addresses of recipients who changed their mailbox addresses, and the data and procedures needed to define the structure of the database, process user queries about local recipients, communicate results and update information.

3. Network Layer: This layer consists of a network directory system which can be either centralized in a single mailer, or distributed in the mailing network. The functions at the network layer of the directory system comprise support of a network-wide query structure and a data model to manipulate information and define its structure, processing of user queries in the various mailers, concurrency control and data integrity, error recovery, and forwarding of control actions. A network directory system consists of: a network directory with mailer system addresses (including the gateway mailer system address) of organizations, groups and system servers; a forwarding directory with forwarding addresses of organizations and groups; and the data and procedures that define the logical and physical structure of the network directory.

4. System Layer: Consists of an internetwork directory system which can be either centralized in a single gateway mailer, or distributed in the various gateway mailers in a federation of computer mail systems or a single computer mail system. The functions and structure of this layer are much like those of the network layer, but on an internetwork basis (Fig. 6).

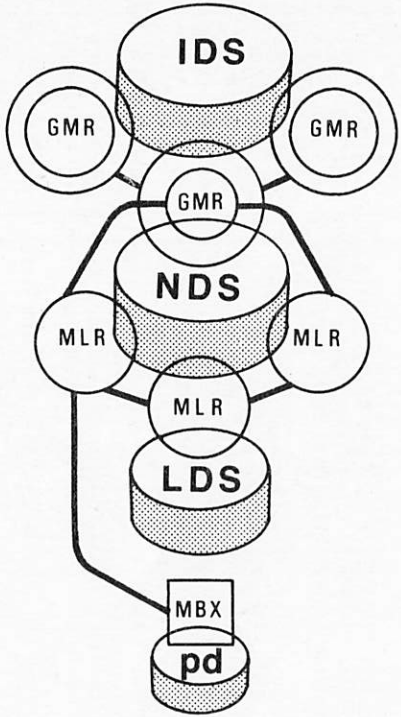
LAYER.	DISTRIBUTION:	FUNCTIONS:
System IDS - Internetwork Directory System		- Forwarding of control actions - Query structure - Distributed Query processing - Data integrity - Error recovery
Network NDS - Network Directory System		- Query structure - Data model mapping (network to internetwork) - Distributed query processing - Concurrency control and Data integrity - Error recovery - Forwarding of control commands
Local LDS - Local Directory System		- Data model mapping (local to network) - Management and description of logical and physical structure of local information - Control of Physical access - Physical storage of detail information
User pd - personal directory		- storage of information about frequently addressed recipients

Figure 6 - DIRECTORY SYSTEM ORGANIZATION

Table 3 - INFORMATION MAINTAINED IN THE DIRECTORY SYSTEM

Layer	Information	Structure	Management
User	Information about frequently addressed recipients, primarily consisting of mailbox addresses	Each user has a user's subschema to describe the information stored in his personal directory A network-wide user-schema is defined for all the users in a mailing network to name the recipients in terms of a standard set of entities, relationships and attributes known to the system (i.e., enter NOLS addresses)	Carried out by user interacting with his mailbox
Local	Detailed information about recipients served by each mailer	Each mailer has a local schema to organize the information stored in its local directory which is independent to the rest of the system	Carried out by mailer in a form completely independent from the rest of the system
Network	Mailer addresses of organizations served in the mailing network	A network schema is defined for all the mailers in the mailing network. Allows for the identification of organizations in the network	Carried out by mailers and the gateway mailer of the mailing network
System	System addresses of organizations served by various mailing networks and other entity of system-wide importance	A system schema is defined for all the gateway mailers in the system. Allows for the identification of mailing networks	Carried out by gateway mailers of the system in cooperation with each other

Structure and Distribution of Information

The logical structure of the information maintained in the directory system is called the schema of the directory system and defines the entities, relationships among entities, and the corresponding attributes, which are of interest to the computer mail system's community. A schema instance consists of all the information which structure is defined by a schema. In our architecture, the information in the directory system has a hierarchical schema formed by four layers (Fig. 6). At each layer, only that information about the entities, and associations between entities of relevance to the layer, is maintained.

Table 3 shows the information maintained at each layer of the directory system, its structure and how the management of this information is carried out. Arbitrary information structures for users (i.e., user subschemas) and organizations (i.e., local schemas) are supported by means of three intermediate data models (Table 3). The most important of those three schemas is the network-wide user schema whose structure determines the format of the NOLS addresses. The system and network schemas support the organization of system-level information necessary to perform the NOLS address-to-mailbox address mapping and the mapping from NEXUS addresses to message routes.

In the directory system we propose, the information is hierarchically distributed by schema instances. That is, at the local layer each mailer maintains all the information about its local users, which constitutes the schema instance of the

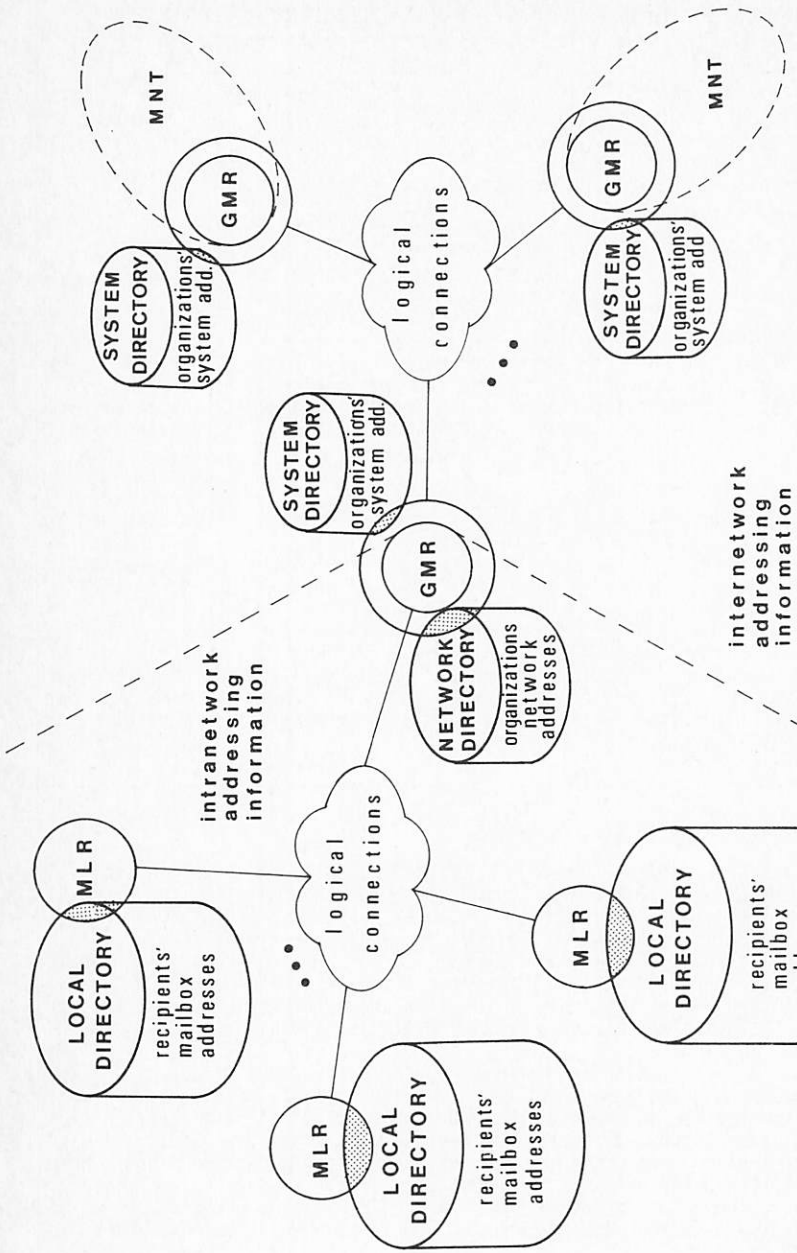


Figure 7 - DISTRIBUTION OF INFORMATION

mailer's local schema. At the network layer, one or all the mailers in the same mailing network store the network schema instance of the network. At the system layer, one or all the gateway mailers store the system schema instance. A distributed network (internetwork) directory system is one where all mailers (gateway mailers) in the network (system) store the network (system) schema instance. When only one mailer (gateway mailer) maintains the schema instance, the directory system is called centralized network (internetwork) directory system. Figure 7 shows the case of a centralized network directory system and a distributed directory system.

The above form of information distribution reduces storage costs because all detailed information is locally maintained at each mailer and only a small fraction of the database (i.e., the network schema instances and the system schema instance) may be redundantly maintained in mailers and gateway mailers (Fig. 7). In addition, this distribution of information also reduces communications costs and the complexity of the communication protocols, as will be clear from the following.

Directory System Control

The control of the directory system basically comprises: distributed query processing, updating of information and error recovery.

As we have pointed out, a NOLS address entered by a sender constitutes a query to the system to obtain the system mailbox address from the data contained in the NOLS address. In this discussion we will refer to IDENTIFY commands (Fig. 5) containing the NOLS addresses created by the senders as user queries.

According to our model, user queries about local recipients are directly resolved by the sender's local mailer, whereas queries about remote recipients are referred to a set of other mailers (Fig. 3). Due to the information obtained from the NOLS addresses, only a small number of remote mailers will have to be queried, on the average, per user query. In addition, since the distribution of information is by schema instances, a user query could very probably be resolved at a single mailer.

A query processing plan (or simply a "plan") is the sequence in which the gateway mailers and/or mailers evaluate a user query. According to the organization of our directory system, there are two possible cases in which a plan is determined. If the network directory system is distributed, the sender's local mailer determines the plan inside its mailing network. This is done by comparing the information in the O, L and S fields of the NOLS address of the sender's message against the information maintained in its network directory. When the user query concerns only local recipients, the mailer compares the NOLS address against the information stored in its local directory to determine (if possible) the desired mailbox system addresses; otherwise the sender's local mailer simply broadcasts the same IDENTIFY command to all the mailers (and/or the gateway mailer).

If the network directory is centralized, the local mailer first determines, based on its local directory, if the query only concerns local recipients; if not, the user query is sent to the mailer with the network directory and that mailer determines the rest of the plan.

The gateway mailers are in charge of the query processing plan at the internetwork level and the transmission of the user query to the appropriate mailers in the remote mailing networks.

Note that a query processing plan may not be known in its entirety by the sender's local mailer if it involves other mailing networks and/or the user query has to be forwarded (Fig. 5). Once the sender's local mailer obtains the replies from all the processes the mailer has queried, it has either established a NEXUS, in

which case the message can be delivered, or the queries have failed to identify the mailbox system address of the recipient(s).

Provision could be made in the mailing protocol [3] to allow for the delivery of a message to mailers instead of mailboxes, so that the receiving mailers can post it on a bulletin board to which all users can have access. In this form messages could be delivered even if the correct mailbox address is not known and/or the sender is only interested in making his message available to a group of recipients in some area and/or related with some organization(s). One or more bulletin boards could be maintained in each mailing network.

The updating of information is also very simple in the proposed directory system. The information stored in the local directories needs no updating at a network level since each local directory is independently maintained by a mailer. The information stored in network directories and system directories has to be updated if the network directory system, or internetwork directory system, is distributed. Since system addresses of groups and organizations rarely change, the updating strategy can be simpled as follows:

> A command asking for the identification of a recipient or the delivery of a message is received in a mailer or gateway mailer. The queried mailer or gateway mailer then determines whether or not the organization specified in the NOLS address is listed in its directories and whether it has moved to another mailer. If it has moved, an UPDATE command is sent back to the sending mailer or gateway mailer (Fig. 5).

The error recovery procedures in our architecture require that the information maintained in local directories must never be completely destroyed, since it is locally maintained by mailers. If information in some network directories or system directories of distributed directory systems is destroyed, a copy of that information can be transmitted without penalty because it represents only a small portion of the database. If a mailer or gateway mailer fails and the information in its network or system directory is out of date (a very rare event due to the fact that only organizational addresses are maintained at the network and system layers) the forwarding and updating capabilities of the system (Fig. 5) can bring the process back into normal operation without disrupting the normal interaction between mailers and gateway mailers.

V. CONCLUSIONS

In this paper we have presented a general description of addressing and delivery techniques for large, distributed computer mail systems with transparent identification services. A hierarchical directory system was proposed to organize the interaction of the various components of the architecture and its importance in the delivery process was discussed.

The provision of identification services in computer mail systems has been often overlooked and very few attempts to present solutions have been made [3], [8], [10]. But the subject is certainly of importance for the establishment of large systems spanning large companies and countries.

It should be emphasized that the concepts presented in this paper constitute a starting point in the development of complete addressing schemes for large computer mail systems. Also, we have proposed our architecture under the assumption of the existence of a communication protocol; the specification of such a protocol is the subject of a forthcoming paper.

REFERENCES

[1] Chu, W.W., Performance of File Directory Systems for Databases in Star and Distributed Networks, AFIPS Conf. Proceedings, Vol. 45 (1976) 577-587.

[2] Feinler, E.J., The Identification Data Base in a Networking Environment, Conference Record 1977 National Telecommunications Conf. (1977) 21:31.1-21:3.5.

[3] Garcia Luna Aceves, J.J., A Study of Computer Mail Services, M.S. Thesis, Dept. of Electrical Engineering, University of Hawaii at Manoa (Aug. 1980).

[4] Henderson, D.A. and Myer, T.H., Issues in Message Technology, Proc. of the Fifth Data Communications Symposium (Sept. 1977), 16.1-16.9.

[5] Kuo, F.F., Message Services in Computer Networks, in: Bauchamp, K.G. (ed.), Interlinking of Computer Networks (D. Reidel Publishing Co., Boston, 1979).

[6] Panko, R.R., The Cost of Computer Mail and Other Media, Working Paper, Proj. 6859, SRI International, Menlo Park, (A94025 (July 1978)).

[7] Panko, R.R., Standards for Computer-Based Message Systems, Report NBS GCR 80-210, National Bureau of Standards (1980).

[8] Panko, R.R., A Survey of Electronic Message Systems, Pacific Telecommunications Conf. Proc. (1981) A3.1-A3.10.

[9] Pickens, J.R., Feinler, E.J. and Mathis, J.E., The NIC Name Server--A Datagram-Based Information Utility, Proc. of the Fourth Berkeley Conf. on Distributed Data Management and Computer Networks (Aug. 1979).

[10] Schoch, J.F., Inter-Network Naming, Addressing and Routing, Proc. COMPCON (1978 Fall) 72-79.

[11] Taylor, H.D., "H.P. Communications Systems" Conf. Record National Telecommunications Conf. (1977) 21:6.1-21:6.5.

[12] White, J.W., Texas Instruments Computer Communication Network and Its Support for the Automated Office, AFIPS Conf. Proc., Vol. 49 (1980) Nat. Comp. Conf., 515-526.

[1]The efforts of one of the authors (Franklin F. Kuo) on this paper was supported by the Office of Naval Research under Contract N00014-78-C-0498.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

INTERCONNECTION OF ELECTRONIC MAIL SYSTEMS - A PROPOSAL ON NAMING, ADDRESSING AND ROUTING

Ian H Kerr

Bell-Northern Research
Ottawa
Canada

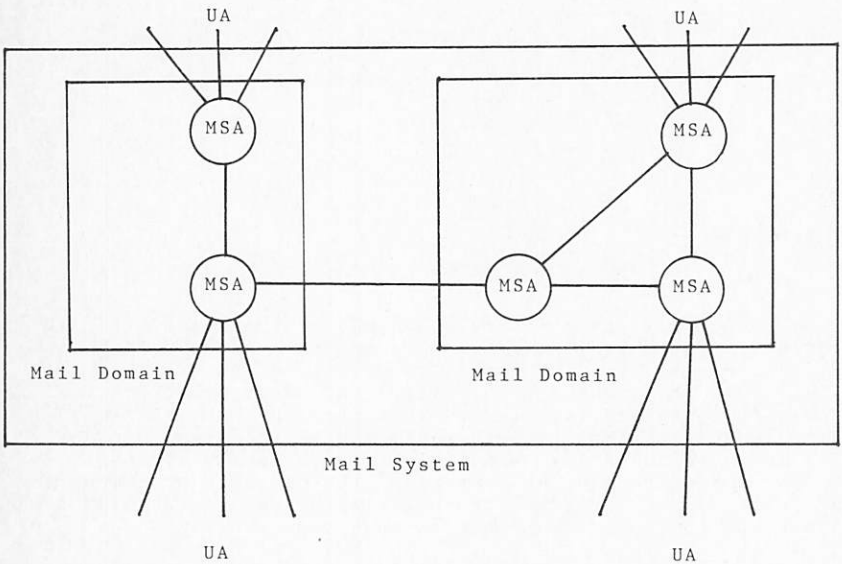
1.0 INTRODUCTION

Many electronic mail systems are planned or already exist. They can provide a very convenient and inexpensive means of communication, and in the future a large proportion of business communication may use this medium. However, the very proliferation of electronic mail systems will seriously reduce the usefulness of the technology unless methods of interconnection are agreed and implemented. The most important aspect of a communication system is the number of people who can be contacted through it. No amount of elegance in the user interface can help if the person one wishes to send a message to is not available through the system.

If electronic mail systems are to be interconnected, standard ways of identifying the intended recipients of the messages must be agreed. Since there are many ways of referring to people, agreement on standards may be difficult to reach. This paper proposes some conventions for identifying individuals, and locating them within a set of interconnected electronic mail systems. The objective is to define sufficient standards to make the interconnection simple, while as far as possible preserving flexibility in the way people are identified and freedom for designers of individual mail systems.

This work is based on the functional model devised by the North American Systems Environment Subgroup of IFIP WG 6.5. <1>. The model is illustrated in Figure 1 and the terms are used as defined in <1> except that 'MSA' (Mail Service Agent) replaces 'ZQL'. In this model, the User Agent or UA is the group of functions which are provided to assist the user in the preparation, examination and storage of messages whereas the MSA is concerned with delivery functions only. For the purpose of this paper the concept of Mail Domain is introduced. A Mail Domain is part of the network of MSAs which comprises the Mail System. Mail Domains are assumed to be managed and owned by separate organizations and correspond to the private and public 'electronic mail systems' referred to above. The IFIP model therefore represents an ideal situation in which the global Mail System has a uniform appearance through the application of International Standards. It is a desirable objective to move towards this situation, but in the meantime the concept of Mail Domains recognizes the fact that different operating authorities will create visible differences within the Mail System. One of the major differences between Mail Domains will be in the way that users are identified. The conventions discussed in this paper are

therefore those which will be necessary for the agreement of protocols for the interconnection of Mail Domains.



MSA = MAIL SERVICE AGENT

UA = USER AGENT

Figure 1. Model of Mail System

2.0 IDENTIFYING MAIL SYSTEM USERS

The concept of a set of attributes which serves to identify an individual for the purpose of sending mail to him is described in <2>. Attributes such as a person's name, job title, department, division and company are pieces of information we know about the person which are independent of any electronic communications system. The most convenient system would allow the use of just that information to address a message. In this paper it is proposed that the attributes themselves appear in the envelope, encoded as a set of type/value pairs, and are used to perform

delivery. There are problems however, especially in the context of interconnected Mail Domains which each have authority over the attribute sets they use. These are:

- o the complexity of the directory access necessary to find the current location of a user from the set of attributes provided.

- o the potentially large number of attributes which could be used to identify an individual. This will make it difficult to reach agreement on a universally implemented set.

The first problem can be restricted by specifying that only a relatively small number of attributes, relevant to normal business communications, can be used. Given this restriction, it is assumed that the cost of the directory accesses can be made acceptable. There are some operational systems which work this way. For example, COCOS <3> uses personal name, department, location and company. TELEMAIL <4> uses personal name, up to four levels of hierarchical structure within a company, and company.

The second problem is much more significant as a lack of agreement on addressing standards will seriously damage the utility of Mail Domain interconnections, although such interconnections would still be possible. Let us suppose that each Mail Domain defines a different set of attributes for identifying its users. e.g., Mail Domain A uses 'name', 'organization name' and Mail Domain B uses 'name', 'department name', 'location'. How can interworking be achieved? Note that although the attribute 'name' is used by both Mail Domains, the acceptable formats of the values of these attributes could be very different. For example, the character set, significance of upper or lower case, acceptable punctuation characters and maximum length could all be different for the same attribute.

Primitive solutions are possible. For example a user on Mail Domain A could be given a second set of attributes for use within Mail Domain B. A gateway process would map from this second set to the set known in Domain A for any message from Domain B addressed to that user. This solution has the obvious disadvantage of requiring the assignment of extra attributes and the maintenance of extra databases, from the service supplier's point of view. From the user's point of view it has the major drawback that a user has several different sets of attributes depending on where a message is sent from. However, this solution does have the advantage of requiring no changes to the user interface or message formats within an existing Mail Domain for the purposes of interconnection.

A second solution is that mail items in all Mail Domains, in addition to carrying the attribute set which they understand and use, can carry an uninterpreted and unrestricted field containing the attribute set to be used in another, interconnected, Domain. It is then up to the originating user to discover what the acceptable set of attributes is in the destination Domain, and enter it (complete with separators) in the message. This solution has the advantage from the service supplier's point of view that the ability to transport these uninterpreted attribute fields, and the ability to carry a message to a connected Domain are the only special functions required to achieve interconnection. From the user's point of view, the solution is more acceptable in that one attribute set, defined in the user's 'home' Mail Domain, is valid

from anywhere. (A way of getting the message to the 'home' Domain must be known. This point is discussed in Section 3 below). However, the user entering a message for a recipient in an alien Mail Domain has a problem in entering the required set of attributes. He needs to find out what attributes are required, in what form they are required, and how they have to be identified and separated from one another. Of course it is always possible that the intended recipient has told the originator exactly what characters have to be entered to ensure delivery of the mail item. However, the intention of the use of attributes is that one should not have to be told specific Mail System addressing information, but should be able to use information obtained by existing means, such as from a business card or letterhead. It will be especially irritating if the user entering an attribute for an alien Mail Domain has to use a different format from that which he normally uses. (For example if I can normally enter 'John Smith' to refer to a recipient of that name, I do not want to have to know that 'jsmith' is required in Mail Domain X, and 'J. Smith' in Mail Domain Y)

For these reasons it is proposed that there is an agreed superset of attribute types which have standard, universally known names, formats and encodings. Particular Mail Domains will internally use a subset of this list, and the subset used will be a well known feature of the Mail Domain.

Any particular attribute will then have the same name and form everywhere and the user should find it easy to address inter Mail Domain messages. In addition, since there will be a limited number of attributes, the user interface in a particular Mail Domain could easily offer the extra service of assisting the user with the input of inter Mail Domain Addresses. This assistance could take the form of prompting for particular attributes, checking the format and inserting the necessary delimiters. The protocol for transferring mail items between Mail Domains will include a standard syntax for the type and value field of each attribute in the superset.

A suggested superset of the attributes which can be used in different Mail Domains is given below.

- Personal Name
- Job Title
- Country
- Company
- Company subdivision 1
- Company subdivision 2
- Company subdivision 3
- Company subdivision 4
- Location

o Project

The attribute types are fairly self-explanatory. The intention of 'Company subdivision 1' etc. is to avoid specifying whether companies are organized into Groups, Operating Units, Divisions or whatever. Then intention is that the subdivision names are filled in numerical order. Thus the address 'J Doe, Widget Dept. Systems Division, ACME Co.' would be specified as:

Personal name = J Doe

Company = ACME

Company subdivision 1 = Systems

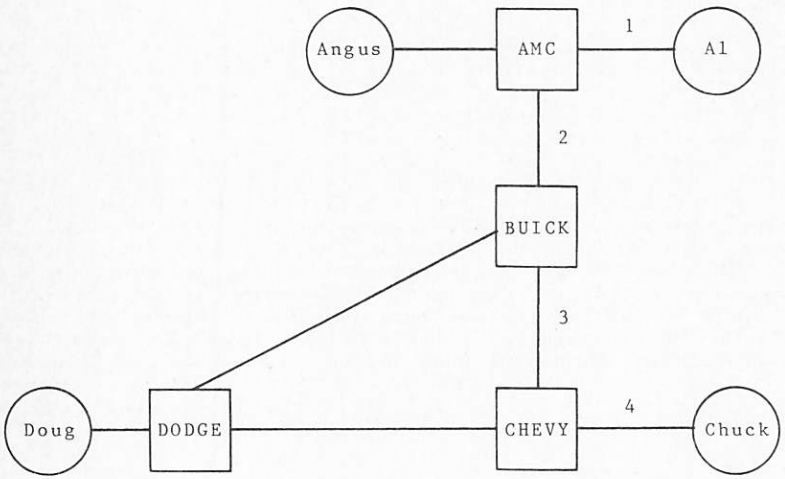
Company subdivision 2 = Widget

Although the above discussion has assumed that different Mail Domains are distinguishable to the user because of their different addressing conventions, it is desirable in the long term to work towards concealing these differences from the user. The existence of a universally agreed set of attributes will assist in this process. Rather than require the user to specify the Mail Domain to which a message is addressed, it is possible just to accept a list of attributes, and for the originator's Mail Domain to pick out those which allow it to determine the Mail Domain of the recipient. For example Mail Domain A could keep a list of what organizations are in fact subscribers of Mail Domain B. A user on Mail Domain A could then just enter a message including organization name in the attribute list without knowing whether the recipient was in Mail Domain A or B, provided that the form of the attribute 'organization name' was common between the two Domains.

3.0 ROUTING

The intention of the attributes defined in the previous section is that by specifying enough of them, the originator of a message creates a globally unique designation of each recipient. Ideally, the Mail System should require no more information in order to decide where a recipient is, and what route the message should take to reach the recipient. However, this ideal situation may not be achieved for some time, and therefore there will be a requirement for the user to provide some information in addition to the set of attributes which will assist in the routing process. The definition of this extra information and the way it is used is discussed in this Section. Three assumptions will be made. One is that Mail Domains as a whole have addresses valid in other Mail Domains to which they are connected. This means that the address of a particular interconnection point (or gateway) need not be specified, but a Mail Domain is free to choose the most suitable interconnection point, and can hide topographical changes from the user. Secondly it is assumed that the network of Mail Domains is not fully interconnected. Therefore one or more intermediate Mail Domains may have to be traversed in the path from source to desti-

nation. This might typically occur when a private Mail Domain is interconnected to a public Mail Domain in one country. The only path from a private Mail Domain in that country to one in another country might involve going via the two national public Mail Domains. This lack of full interconnection may come about because of regulatory reasons or because of the lack of agreed protocols or physical interconnection facilities between each pair of Mail Domains. A further assumption is that routing within one Mail Domain is invisible to the user, and is accomplished entirely by means of the attributes understood within that Mail Domain.



AMC, BUICK, CHEVY and DODGE are Mail Domains

Al, Angus, Chuck and Doug are Users/User Agents

Figure 2. Example of message routing

The following discussion uses the configuration shown in Figure 2 to illustrate some of the routing and addressing issues. Al, Angus, Chuck and Doug are users, identified by the attribute sets A (Al), A (Angus), A (Chuck) and A (Doug). AMC, Buick, Chevy and Dodge are Mail Domains, and their names are considered to be sufficient identification.

Consider a message from A1 to Chuck. The various possibilties for the information required and the corresponding routing techniques are:

- Attribute set only.

  A1 supplies 'A (Chuck)' and AMC is able to use some of the attributes from the set provided to decide to hand the message on to Buick. Buick uses some attributes to decide to hand the message on to Chevy. Chevy uses the remaining attributes to locate Chuck. As mentioned earlier, this technique is the best from the user's point of view, as he does not need to know anything about the Mail Domains or their connectivity. However, it will in general be difficult for the necessary knowledge about attributes to be available at the right place as there would be a considerable effort in setting up and maintaining the necessary directories. Although this technique might be achieved in the long term, the MSA to MSA protocol should not rely on it.

- Final Mail Domain name plus attribute set.

  A1 supplies 'Chevy; A (Chuck)' where the attribute set 'A (Chuck)' is only interpreted within the final Mail Domain. This technique requires that Mail Domains have names that are globally unique, and that any Mail Domain is able to select the next point in the route towards the named final Mail Domain. Some Mail Domains may conform to a naming standard which ensures uniqueness (e.g. a CCITT Recommendation in the case of carrier provided Mail Domains). Otherwise, a globally unique name can be formed by adding the owning company name (presumed to be unique within a country) and the country name to the common name of the Mail Domain. However, even if unique names exist, it will be a problem for any Mail Domain to maintain a routing table with entries for all other Mail Domains.

- Complete route plus attribute set.

  'Complete route' means the ordered set of Mail Domain names defining the path from source to destination. In the example, A1 supplies 'Buick; Chevy; A (Chuck)'. This approach simplifies the requirements on the Mail Domains at the expense of requiring the users to know the detailed topography of the Mail System. Mail Domain names no longer have to be globally unique, as the only requirement is that a Mail Domain recognises the names of those Mail Domains to which it is directly connected. However, unique Mail Domain names will still be desirable for other reasons, such as the generation of unique message identifiers. Since the Mail Domain must know about the Mail Domains to which it is directly connected in any case, the problem of maintaining routing tables is much reduced with respect to the previous solution.

The three techniques outlined above are simplified and are not intended to exclude the posibility that more complex actions are taken by the Mail Domains. These might include;

- A Mail Domain polls other Mail Domains to find one which recognises a given recipient rather than maintaining full routing tables.

- A Mail Domain overrides the route or part of the route specified by the originator if it knows of a better way to reach the recipient.

It seems unwise at this stage to preclude any of the above routing techniques by placing restrictions in an interconnection protocol. To allow any of the possibilities all that is required is that the attribute set which identifies the recipient may be accompanied by one or more Mail Domain names which are used to assist in routing the message to the recipient. This set of information (i.e. attribute set plus Mail Domain names) is referred to as the 'designation' of the recipient.

If the designation of a recipient is allowed to contain some routing information, then it immediately follows that a particular designation will not be valid everywhere. In fact this problem is not confined to the case where Mail Domain names are included in the designation. The Mail System may accept a set of attributes which by itself is not sufficient because it assumes that if any attributes are missing from the designation of a recipient, the values which apply to the originator can be used. Thus I may not need to specify 'company' when sending a message to someone who is an employee of the same company, for example.

The problem arises because the recipient of a message will at least want the ability to reply to the originator, and probably also to 'reply' to the other recipients. If this is to be possible, the recipient must be provided with, or be able to work out, designations for the originator and all other recipients which are valid from where he sits. This problem is fully discussed, in a different context, in <5>.

From the example on Figure 2 , suppose that the complete route must be provided by the user and that A1 sends a message to Angus, Doug and Chuck. At posting (i.e. at the point indicated by '1' on the figure), the envelope will contain the following designations:

RECIPIENTS: A (Angus)

Buick; Dodge; A (Doug)

Buick; Chevy; A (Chuck)

ORIGINATOR: A (A1)

There are two possible policies for how these designations are handled in order to meet the objectives defined above.

The first is that all designations are kept valid at any point in the route. Considering the copy destined for Chuck. When the message is passed from AMC to Buick (2) the designations would be:

RECIPIENTS: AMC; A (Angus)

Dodge; A (Doug)

Chevy; A (Chuck)

ORIGINATOR: AMC; A (A1)

When the message is passed from Buick to Chevy (3) the designations would be:

RECIPIENTS: Buick; AMC; A (Angus)

Buick; Dodge; A (Doug)

A (Chuck)

ORIGINATOR: Buick; AMC; A (Al)

The designations can then be passed unchanged to Chuck on delivery of the message (4) and will be valid if Chuck uses them in a reply or another message.

The actions of any Mail Domain (A) under this policy can be summarized as follows. A copy of a message is handed off to another Mail Domain (B) in order to be delivered to a number of recipients. Remove the name of Mail Domain A from the designations of these recipients, and prefix the name of Mail Domain A to the designations of all other recipients.

The second policy is simply to ensure that the designation of the originator is valid on delivery of the message, as valid designations for all other recipients can then be worked out. It is an open question whether the MSA or UA should have the responsibility for generating the valid designations. However, for the purpose of the MSA to MSA protocol it is sufficient to ensure that the last Mail Domain has the necessary information. Therefore, when the message for Chuck is passed from Buick to Chevy (3) it will contain the following designations:

RECIPIENTS: A(Angus)

Buick; Dodge; A (Doug)

Buick; Chevy; A (Chuck)

ORIGINATOR: Buick; AMC; A (Al)

By comparing the route back to the originator with the route from the originator to the other recipients it is easy to generate routes from a recipient to the other recipients. There are a number of advantages to this policy. Firstly, the valid designations need not be worked out unless the recipient really does want to reply. If this is relatively infrequent, some effort is saved. Secondly, there is no need to add extra fields to the recipients' designations during delivery of a message, which will save some processing effort. Of course, extra fields will need to be added to the originators' designation, but there is only one originator, and space could be reserved, so that re-formating of the envelope is avoided. Thirdly if, as indicated in the example, there is a direct connection between Chevy and Dodge, there was never any point in Buick generating the route 'Chevy, Buick, Dodge, A (Doug)' as Chevy would recognize 'Dodge' and use the direct connection Chevy - Dodge.

Note that in this case it is important not to remove any of the information supplied by the originator as it may all be needed to generate valid designations at the recipients' location.

Therefore, the following assumption should be made about the designations carried in the MSA to MSA protocol. Only the designation of the originator is valid at any point. Valid designations of the other recipients should be generated from the designations provided (which are as provided by the originator) and the designation of the originator himself. The same assumptions should be made about the attribute sets i.e. that the attribute set of the originator is complete and valid but that those of the other recipients may be incomplete and, if so, the necessary values should be taken from the originator's attribute set.

4.0 ENVELOPE AND CONTENTS

In line with the terminology used in the IFIP group, all elements of information required by the Mail System are in an 'envelope' part of a message, and all other information is in the 'contents'. In Section 3 above, the envelope contains a list of recipients, and an originator. This is obviously similar to the 'TO' and 'FROM' fields normally found in business memos. In fact some electronic mail systems use the same list of addresses to provide both the delivery function and the 'TO' and 'FROM' fields. There are a number of reasons why these two sets of names/addresses may not be the same.

- However human oriented we make the Mail System designations, by means of attributes, they may still not correspond to what we want to put in the 'TO' field of a memo. For instance the mail system attribute set may be 'Bloggs', 'XYZ Dept.', 'ABC Corp', but I just want to put 'Fred' in the 'TO' field.

- There are various distinctions between fields in a memo which have no effect on the delivery of mail items. (e.g. 'TO', 'CC', 'File Copy', 'Info only')

- The recipients of a message may be known by a group name which they all recognise (e.g. 'all members of design group'). Therefore the 'TO' field need contain only the group name, whereas the mail item envelope must contain each recipient's designation.

Of course it would be possible to try to cover all these requirements in the list of addresses in the envelope, by means of extra information associated with each designation. However, it would be better to avoid linking agreements on addressing and routing to totally unrelated topics of information management and business practice.

Therefore the envelope should contain a list of recipient attribute sets (and, if necessary, routes) only. The definition of a memo 'TO' field etc. is left to the User Agents, as is any possibility for automatic generation of 'TO' field from recipient list or vice versa. This might be done to avoid the user having to enter the same information twice in case the mail System designations are acceptable for the 'TO' field. It is envisaged that, on receipt of

the mail item, a user has the option of examining the envelope, just as he might look at a conventional mail envelope. Therefore he is not necessarily forced to look at a list of mail system designations as well as the 'TO' field from the contents.

5.0 CONCLUSIONS

For the purpose of interconnecting electronic mail systems, various possibilities for reaching the necessary agreements on names, addresses and routes have been discussed. The resulting proposals are summarised as follows:

- Messages have an envelope which contains information related to delivery. In particular the envelope contains a list of recipient designations and an originator designation.

- Designations consist of a set of attribute type/value pairs and, if necessary, one or more Mail Domain names which are used to assist in routing the message.

- The set of attributes used to identify a user is a subset of a global list whose types and encoding formats are standardised. The subset required is defined by the Mail Domain to which the message is to be sent, plus possibly some defined by intermediate Mail Domains.

- Mail Domains have globally unique names either by conforming to a standard, or by including the country name and the owning company name in the Mail Domain name.

- When a message is transferred from one Mail Domain to another, the designation of the originator is guaranteed to be valid. The designations of the recipients to whom this copy of the message is not to be delivered are as supplied by the originator and may need elements of the originator's designation to be added in order to make them valid.

- The coding of the envelope does not attempt to satisfy the requirements of business memo headers. If necessary, a memo header should be carried separate from the envelope.

6.0 ACKNOWLEDGEMENTS

The author wishes to acknowledge helpful discussions with Ian Cunningham, Peter Schicker, Sabine Rohlfs and Hilary Williamson, as well as the other members of IFIP W.G. 6.5 North American Systems Environment subgroup.

7.0 REFERENCES

<1> Proceedings of the IFIP W.G. 6.5 St. Augustin Workshop on Computer Message Systems, May 1980

<2> Naming and Addressing in a Computer Based Mail Environment, P. Schicker, to be published in Computer Networks

<3> The Design and Service Impact of COCOS - An Electronic Office System. N. Dawes, S. Harris, M. Magoon, S. Maveety, D. Petty. This Symposium.

<4> TELEMAIL Reference Manual. GTE Telenet, August 1980.

<5> A Network Independent Transport Service, The Post Office PSS User Forum, February 1980

COMPUTER MESSAGE SYSTEM IMPLEMENTATION

Chairman - P. Gupta

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

MSG: A SIMPLE MESSAGE SYSTEM

John Vittal

Bolt Beranek and Newman Inc.
50 Moulton St.
Cambridge, MA 02238

Electronic mail systems have been in active use for many years. This paper describes one of the earliest successful network-based messaging systems, MSG, discussing the precepts for its development, the functionality that it provides, and its user-interface. Some of the reasons for its success are noted, especially the benefits of user feedback during the development process. It is also compared to other message systems.

INTRODUCTION AND HISTORY

Electronic mail systems have existed for many years. At the simplest level, electronic mail systems transmit messages (a unit of communication) between originators and recipients. Often, a computer mediates the process, with the resultant system being called a computer-based message system (CBMS). Computers serve both to mediate the actual communications between systems and to provide users with facilities for reading and creating messages.

The notion of a network-wide electronic mail delivery service for computer networks has often been described in the literature [6, 7, 9, 10, 15]. Various user-interface computer programs which may or may not interact with that service to provide electronic mail facilities have also been described [4, 8, 11, 16]. The way users create messages or process incoming mail using such a system may be completely separated from the mechanism that delivers mail. More and more of these interface programs are being developed, both by the research and industrial communities.

One of the earliest successful computer-based message systems is MSG [16]. It was developed for the ARPANet environment [3, 14] in 1974. Its background and history are well worth recording due to its popularity and influence on later systems and how it distinguishes itself from other systems of the time.

As the ARPANet was initially being developed in the early 1970's, Ray Tomlinson of Bolt Beranek and Newman Inc. (BBN) created two programs to run on the TENEX time sharing operating system [2]. One, called SNDMSG, allowed users to create messages and interfaced to the delivery mechanism. The other, READMAIL, allowed users to print out "recent" mail[1] on their terminal.

However, these programs did not permit the kind of message processing that was required by high volume traffic, at the time

considered to be approximately ten messages a day. Larry Roberts, then the director of the ARPA Information Processing Techniques Office, built a message printing program called RD which allowed users to survey the mail in their inbox and to selectively print the messages from that computer-based file. RD was built using a programmable text editor, TECO [12].

While the use of these three systems has declined in favor of more powerful systems, their designs have strongly influenced many later systems.

In about 1972, subsequent to the development of RD, Barry Wessler of Telenet wrote a program called NRD (New RD). He intended it to be a successor to RD, but never publicly distributed. In addition to the functionality provided by RD, it added the ability to manage more than one file of messages by providing some filing, retrieval and cleanup[2] functions.

In 1973, Martin Yonke, at the time associated with USC's Information Sciences Institute (ISI), obtained a version of NRD. With the author of this paper, he added more functionality, provided a uniform user-interface to all commands, and coupled-in the message creation program, SNDMSG. The result was first named WRD and later called BANANARD [17]. It was the first CBMS in the ARPANet community which integrated the message reading and creation functions by providing a single, friendly user-interface that included a command driver and on-line help facility for all commands.

Simultaneously with the development of BANANARD, in early 1974, this author - then also associated with ISI - developed MSG. It was based on an early version of BANANARD, to which functionality and a slightly different user-interface were added. The primary functions that MSG's enhancements facilitated were the ability to forward messages to other people and to reply to mail. It also allowed users to tailor aspects of MSG's behavior through the use of some commands and a user profile. Although MSG and BANANARD provide an integrated environment for message creation and reading, SNDMSG is used for all mail composition and sending functions.

MSG is widely disseminated, being available on over twenty TENEX and DECSYSTEM-20 systems. At one time, it was the system of choice for more than a thousand users, and is still in active use throughout the ARPA community and elsewhere. In addition, its design has been the basis of several more recent CBMSs, and its user-interface has been copied by at least three PDP-11 based message systems, also named MSG.

Since MSG was developed, many new CBMSs have been written for a variety of hardware and contexts. One of the more popular and publicized ones, Hermes [8], addressed the same user population as MSG. However, the early availability of BANANARD and MSG seemed to provide a catalyst to CBMS use by crossing a threshold for functionality and usability not before reached in a CBMS. The addition of the reply function made MSG the first to close the circle by providing a "complete" set of functions for message processing.

The rest of this paper discusses the precepts behind the development of MSG, its user-interface and the functions it provides. It is

compared to the other systems available at the time it was built. Some of the reasons for its success are discussed.

A DESCRIPTION OF MSG

Precepts

For the purposes of this paper, messages are textual entities communicated between people and/or computer processes. A message consists of a set of named fields; a field has a specific semantic meaning. The format of the messages sent over the ARPANet is described in a "Standard for the Format of ARPA Network Text Messages" [5]. CBMSs use fields to indicate to whom the message is being sent (the To and Cc fields), a topic for the message (the Subject field), and the body of the message. All of the fields of a message, other than the body, are considered to be the "headers" of the message. In addition to these fields, messages have some ancillary properties such as length, delivery date, and status indicators such as whether the message has been examined by its recipient or whether the message has been deleted. Most CBMSs do not physically remove a message which is marked as deleted from the mailbox until cleanup is requested by the user.

The intent of MSG and BANANARD was to provide a simple yet powerful system, concentrating on the user functions for reading and filing messages, and using existing tools for message creation. The primary requirement was for a good user-interface to a set of integrated and consistent simple functions.

To this end, a user-interface characterized by single letter commands was built. It was expected to interface to a small number of commands. Because of user feedback, the functionality and user-interface of MSG changed over time. Originally, the functionality was very similar to that of BANANARD.

User-Interface

At the time it was built, MSG was relatively novel in providing an integrated help facility through which users could obtain a description of the meaning and form of commands and their arguments. A profile was also provided to allow users to modify the amount of information (and prompting) which the system provided; experienced users need less prompting than novice users.

MSG's user-interface is deceptively simple. It is designed to be used with printing terminals. The first character of all of the command names is all that is needed to specify the command. The rest of the command name and the prompting for its first argument, if it takes one, are automatically printed on the user's terminal. A similar style is used, where possible, for additional arguments or confirmation of the commands. The amount of prompting information can be controlled directly by the user, or indirectly through the use of the user profile. Command confirmation, either explicit or implicit, is required only by those commands which might cause disastrous effects. An implicit confirmation is given when, for example, a file name is uniquely specified (by typing a carriage return), and the file name argument is the last argument that the function takes. Some users, however, wanted to be able to explicitly confirm all commands. This capability was added and is

specifiable through the use of the profile.

MSG has a small number of commands. For more complex systems, this style of user-interface presents problems. It allows only a limited number of commands without having to go to special, non-alphabetic characters for the command name. Also, the mnemonics of many commands tend to be contrived, since the name that users would prefer begins with a character already in use.

The final aspect of the user-interface is the use of exactly one mechanism for aborting commands. The mechanism may be used either during command specification or execution. In other words, it allows users to have a uniform mechanism for indicating that they wish the system to "stop what its doing" and return to the command level. MSG was one of the first user programs to include this feature. This is now recognized to be a critical feature of both operating system executives and user programs.

Functions provided

The commands that MSG provides can be organized into several categories. The following is a list of the functions that MSG provides, by category, with a very short description of what each command does. A more extensive description of these functions and their arguments is given in Appendix A.

Viewing	Backup	Print the previous message
	Headers	Print a summary of message headers
	List	Print messages on line printer
	Next	Print the next message
	Type	Print messages
Creation	Answer	Replies to a specific message
	Forward	Sends messages to others
	Sndmsg	Calls the SNDMSG program
Motion	Move	Copies messages into a file and marks them as deleted
	Put	Copies messages into a file
File operations		
	Delete	Marks messages as deleted
	Exit	Cleans up file and exits MSG
	Overwrite	Cleans up file
	Read	Inputs a new mail file
	Undelete	Marks messages as undeleted
	Write	Sorts a file by message arrival date
User-interface and Profile		
	Koncise	Concise typeout mode.
	Include length	Includes the length of messages in the headers listing
	Verbose	Verbose typeout mode
	Zap Profile	Goes into the profile dialog
Marking	Mark	Marks messages as examined
	Unmark	Marks messages as not examined
Help	?	Describes a command and its arguments

Miscellaneous

Current message	Provides status information
Exec	Gives the user a new layer of the Tenex executive
Go to	Sets the current message number
Jump	Lets the user run another program
Print date	Prints the day, date, and time
Quit	Exits MSG
Comment	Gathers text

Messages which are printed in their entirety (with any Viewing command except Headers) are marked as having been "examined".

The most common type of command argument is that of a "message sequence". A message sequence is a set of messages from the currently active message file to which MSG is to apply a specific operation. See Appendix B for a more complete description of how to specify a message sequence.

The only other primary argument type is a file name. This is a specification for a file, and an operating system call is used to obtain the file name from the user.

User Profile

The user profile permits relatively permanent modification to MSG's normal (default) behavior (both functionality and user-interface). It allows a user to specialize a system to his/her own needs.

As a system becomes more complex, the user profile for that system needs to become more robust to provide its users with an adequate level of flexibility. Profile mechanisms for a system can become highly robust regardless of the complexity of the system, allowing user modifications and additions to the functionality, new commands, and complex modifications to the user-interface.

A highly robust profile mechanism could have been built for MSG, but there were advantages to keeping it simple. One user may often find it convenient to use another's profile while interacting with a system. This can cause problems since a system which has been highly specialized by one user via the use of the profile makes that modified system difficult to use effectively and very confusing to other users. This confusion can come because the system has been specialized to the point that it does not match either the manual or the user-model[3] of the default system, much less the system as modified by another user's profile. So, robust modifications to a system might cause the system to act differently, perhaps even as an entirely different system. The simple profile used by MSG does not allow complex modifications to the system and therefore does not fall into this trap.

Another advantage is that profile mechanisms which allow only simple changes to system functionality are generally simple and easy for users to comprehend and use. Sometimes with a complex profile mechanism, a completely separate user-interface and user language is required. This forces the user to understand two "systems" rather than just one: one providing the functionality and the other allowing modifications to that functionality.

In keeping with the philosophy of simplicity, the MSG user's profile

mechanism (reached using the "Zap profile" command) takes the user through a dialog, setting various controls on the amount of prompting and data the system will output in various circumstances. A sample of the kinds of questions asked users, which allows them to vary system behavior, are:

- Include the length of each message in the headers listing?
- Check for new messages while not actively interacting with MSG?
- List the headers of messages being deleted (by commands 'Delete' or 'Move')?
- Always confirm all commands with a single carriage return?
- Do you want to see what gets passed to SNDMSG by the Answer command?
- Do you want to receive copies of your 'Answers' to messages?

The user-interface to the profile mechanism is the same as the standard command language: single character with automatic completion. A question is asked, the current and default settings printed, and a new value requested.

How MSG Changed Over Time

User feedback was one of the most important aspects of MSG's active development. Not only did the functionality of the system change as a result of this feedback, but so did the user-interface. This feedback provided a means of testing the utility of many features and user-interface behaviors. It allowed the system to be what the majority of users desired rather that what the designer of the system felt the users should have.

MSG started from a set of primitive message processing operations. Several of the commands listed above were not implemented in the initial version of MSG:

- Creation: Answer and Forward
- Motion: Move
- File operations: Write
- Marking: Mark and Unmark
- User-interface and Profile: Koncise, Verbose, and Zap profile
- Miscellaneous: Print date and Comment

It became clear, even before MSG was first publicly released, that the operations of Put and Delete were so commonly used together that a combining operation (Move) should be included in the functionality of the system. This was the first major modification.

It also became obvious early on that most users who were familiar with the system required less prompting than inexperienced users. At this point, the Koncise and Verbose commands were added to reduce or expand the amount of prompting information presented to users respectively, depending on their estimation of their own level of

expertise. Shortly thereafter, the profile mechanism reached by the Zap profile command was introduced, initially as a means of making standard on a per user basis the amount of prompting information that was desired.

Another addition, made soon after the public release of MSG, was the message creation interface realized by the Forward and Answer commands. The Forward command effectively includes copies of a set of messages into the body of a message being created, passing this information to SNDMSG. The rest of the data required to send the message, specifically the address or any additional information that the user might want to add to the body of the message, were gathered by SNDMSG itself.

The Answer command was similar. It looked at a message for the addresses to which a message's reply should be sent. It passed these, as well as the contents of the original Subject field of the message, to SNDMSG. The body of the message was gathered by SNDMSG.

The Answer command was one of the more complex, but the most important, additions to MSG. Not only was it the largest implementation task of any command in MSG, but given its complexity and number of alternatives, it became the hardest command to "get right" in the user-interface. For this reason, it was not included in the initial version. The current implementation of the operation was the third one developed, being the conceptually simplest and least invasive of the interfaces. It also was the one with the fewest options.

These two commands were the most difficult to provide given the decision to use SNDMSG as the message creation tool. However, the inclusion of them caused MSG to cross the threshold of utility and become accepted and used by a large number of users. Many users left the paradigm where they used separate tools for message reading and creation only after these functions became available.

The remaining modifications were the least important. The marking functions were included for two reasons. One was to negate the effects of typing or listing a message in order to highlight messages that needed attention in subsequent sessions without adding a completely new mechanism, new commands, and more confusion for users. The second reason was to mark as examined messages that a user did not want to see. In other words, it was intended to fool the system into thinking that the message had been examined completely although it had not been. The rest of the functions were added primarily for user convenience, most often after several users had requested the function.

COMPARISON WITH OTHER SYSTEMS

Many of the other CBMSs of the time have already been alluded to. The very simple systems (SNDMSG, RD and READMAIL) did not integrate the reading and creation functions, had different user-interfaces, and did not provide sufficient functionality for simple message processing.

On the other hand, two systems came very close to MSG. BANANARD gained acceptance, but seemed to not have the right functionality. The user-interface seemed to be a little too verbose for experienced

users. However, it is important to note that some users still prefer to use BANANARD. These tend to be users who view mail rather than respond to it.

HG is a system built by Jim Calvin of BBN. It has about the same functionality as MSG, but is significantly more responsive. One drawback is that new mail can not be delivered during a user session with HG. It was never distributed outside of BBN.

At about the time that MSG was initially being distributed, the Hermes project [8, 9] at BBN was getting underway. Hermes has all of the right things, and more. It has the functionality and a help mechanism. It has a friendly user-interface that was consistent with the TENEX executive's user-interface in terms of how users interact with the system. It has an integrated and uniform message composition mechanism. But, it was developed with the attitude that large volume message processing required a highly flexible and simple data-base management system.

Hermes is a large, complex system, currently with 55 top-level commands and 38 commands for message composition. MSG, on the other hand, has only 30 commands. Hermes also has various kinds of "objects" which drive the message handling process, and editors for each of them. Each editor has its own set of commands. Describing them is beyond the scope of this paper. Hermes has gone to the point of providing a data-management system and a preliminary version of message processing language and environment. This takes Hermes out of the realm of a simple message processing system. Some users want and need the power and flexibility of a system like Hermes. Most, however, don't seem to use all of its capabilities, or use a majority of them infrequently.

MS [4] is somewhere between MSG and Hermes in functionality. It was developed to run on PDP-11 hardware using the Unix operating system [13], and has several user-interfaces. One is an emulation of the MSG user-interface. The other major one more directly approximates that of the Unix shell.

MM, for Tenex, was developed in response to the problem that MSG was no longer developing and did not provide the right level of message creation functionality. SNDMSG, the program MSG uses for all message creation functions, does not permit the editing of any field other than the text field, and then only while it is being specified.

MH [1], for Unix, was done in response to the problem that MS was too robust a system for the resources provided by a time-shared PDP-11 running Unix.

CONCLUSIONS

One adage that can be applied is that, while simple systems are fairly easy to develop, complex systems are much harder to build, debug and "get right". MSG was designed to be simple, easy to learn, relatively self-documenting, and very easy to use. However, it would not have been so successful without the high level of user feedback during the development stage. Some of its predecessors did not provide enough functionality or a uniform user-interface. Some of its contemporaries and successors have gone too far, providing

complex data base management facilities rather than simple message processing functions. It is worth noting that, for some, even MSG is too complex; they prefer a system on the level of BANANARD. One of the key issues, though, is that both MSG and BANANARD are "friendly".

However good MSG is, it is not perfect. Its major drawback is that it does not have a directly integrated message creation facility with the same style of user-interface as the rest of MSG. The result is that users are forced to use two separate interfaces for a single conceptual process -- dealing with mail. In addition, the decision to use SNDMSG limits users because it has no way to edit various fields of the message after a specific field has been completely specified, especially address lists.

Another possible drawback is that of the command language. As has already been discussed, single character command invocation eventually forces system developers into using non-mnemonic commands for an operation (e.g., Koncise) and may limit the number of commands available. If there is a large number of commands, there tends to be a correspondingly large amount of "conceptual load" on the user. One must remember the commands, what each does specifically, and how to get out of "bad" situations. It is much harder for users to track the changes in a developing system if the user-interface to the system is complex.

Many users have indicated that they have switched from MSG to something a little more robust, generally either Hermes or MM. However, in the same breath, they also say that whenever they have a lot of work to do, they prefer to use a very simple and clean system; when this happens, they always revert to MSG. This suggests an approach to the development of robust systems. Develop a simple "kernel" which permits easy and efficient performance of the most common functions and provides a mechanism for "closed-loop" processing (this occurred in MSG with the addition of the Answer command). Then, add easily distinguishable "layers" of functionality for more esoteric work.

However, notifying users of new changes to a system, training them to use the new features, and having them remember the features is a difficult problem. One of the reasons that some features in MSG (for example, multiple message sequences - see Appendix B) were not used was the features were not immediately perceived as useful when announced, and as a result, were quickly forgotten.

APPENDIX A: MSG COMMANDS

Rather than give a long description of all of the commands available to users of MSG, the following provides a summary of their actions. It is an annotated listing from the on-line help information. The "command character" is that which the user types, the rest of the information is what the user sees on his terminal. Data enclosed in angle brackets are either context-dependent output (if lower case) or a request for further user input (upper case). A description of the <MSG-SEQUENCE> argument appears in the following appendix.

Command Character | Meaning

A — Answer message number: <MESSAGE-NUMBER>
Reply to whom the message is:

F	From
T	To: list plus originator
C	Cc: list plus To: list plus originator

Replies to a specific message, filling in the addresses and subject fields.

B — Backing up -- previous message is:

Prints the previous message on the user's terminal.

C — Current message is <current message> of <number> messages in file: <file name>

Provides status information about the current message file, how many messages are in the file, and current message within the file.

D — Delete (message sequence) <MSG-SEQUENCE>

Marks messages as deleted. Messages are not physically removed from the mail file until a cleanup operation has been performed.

^E — Exec [confirm]

Gives the user a new layer of the Tenex executive.

E — Exit and update old file <file name> [confirm]

Cleans up file by physically removing all messages marked deleted and exits MSG.

F — Forward (message sequence) <MSG-SEQUENCE>

Sends messages to others, filling in the body and the subject field. Users may add comments to the body.

G — Go to message number: <MESSAGE-NUMBER>

Sets the current message number.

H — Headers (message sequence) <MSG-SEQUENCE>

Prints a summary of the headers of a set of messages on the terminal.

I — Inclusion of length in header

Switch indicating whether the Headers command includes the length of messages in the listing

J Jump into program: <PROGRAM NAME> [confirm]

Lets the user run another program while not modifying MSG's running environment. When the program is exited, the user will be left in MSG in the same state as when the program was started.

K Koncise -- provides shorter prompting

Switch into or out of concise typeout mode.

L List (message sequence) <MSG-SEQUENCE>
on file name: <FILE-NAME>

Prints messages on some file (generally a line printer). The printout is preceded by a headers summary.

M Move (message sequence) <MSG-SEQUENCE>
into file name: <FILE-NAME>

Puts a copy of messages into a file and marks them as deleted in the current file.

N Next message is:

Prints the next message on the user's terminal.

O Overwrite old file <file name> [confirm]

Cleans up file by removing all deleted messages.

P Put (message sequence) <MSG-SEQUENCE>
into file name: <FILE-NAME>

Places a copy of messages into a file.

Q Quit [confirm]

Exits MSG.

R Read file name: <FILE-NAME>

Inputs a new mail file.

S Sndmsg [confirm]

Calls the SNDMSG program to create and send a new message.

T Type (message sequence) <MSG-SEQUENCE>

Prints messages on the user's terminal.

U Undelete (message sequence) <MSG-SEQUENCE>

Marks as undeleted messages which have been previously marked as deleted.

V Verbose -- provides more prompting

Switch into or out of verbose typeout mode.

W Write file <file name> sorted by message arrival time [confirm]

Sorts a file by message arrival date, also cleaning up the file.

Z Zap profile [confirm]

Enters the profile dialog.

´ Mark messages as ´examined´ (message sequence) <MSG-SEQUENCE>

Marks messages as examined.

- Unmark messages to be NOT ´examined´ (message sequence) <MSG-SEQUENCE>

Marks messages as not examined.

: (the day, date and time are then printed)

? Type command character for its description, ? for summary:

Prints on the user´s terminal a description of the meaning and form of a command and its arguments.

; Comment

Just gathers all user input up to a ^Z.

APPENDIX B: MESSAGE SEQUENCE SPECIFICATION

Message sequences (that is, instances of a <MSG-SEQUENCE>) are specified using one or a combination of the following:

1. Any single message number
2. Message number range (i.e. number <separator> number, where <separator> is ">" or ":"). "15:18" means 15 through 18, but "18:15" means 15 through 18 in INVERSE order.
3. number "-" number. For example, "24-7" means messages 24 through 27, but "24-2" is an error.
4. A sequence of types 1, 2, or 3 separated by commas.
5. Special sequences (must be first character of sequence)

Sequence Character	Meaning
<escape>	current message number
<tab>	last sequence specified by types 1, 2 or 3
A	all messages
D	deleted messages
F	string search of from field
I	inverse order

O	old messages only
R	recent messages only
S	string search of subject field
U	undeleted messages
E	examined (seen) messages
N	Not examined (not seen) messages
L	Last message sequence completely specified
M	Multiple message sequence (described below)

The specification of a multiple message sequence will allow the use of any of the above, with the result being added to a list of messages which will make up the sequence. Carriage return terminates specification. Typing "P" or "+" as the first character will allow the user to add to the list, "M" or "-" will allow subtraction from the list, otherwise the intersection of what is being specified and the current result will be taken. Initially, the ´current result´ is equivalent to ´All messages´. Typing "Q" or control-N (^N) will abort back to the top level of MSG. Typing "A" (for All messages) will reset the state of the multiple message sequence to be "All messages".

Although the multiple message sequence is a powerful facility for specifying message sequences in large files of messages, few people use it. This is primarily because it was added late in the release cycle of MSG.

ACKNOWLEDGEMENTS

The development of MSG was funded, in part, by a contract from DARPA IPTO to USC Information Sciences Institute.

I would like to thank many people for their support and feedback during the development of MSG, especially Steve Walker (at the time with ARPA), Martin Kay and Ron Kaplan of Xerox, Martin Yonke who was with ISI at the time, and Dale Russell and Lee Richardson of ISI. Without their constant encouragement, MSG would not have been developed either in the time frame or to the level that it was.

Many people have been applying pressure for over five years for this paper to be written: Martin Kay, Dan Bobrow, Jerry Elkind, Steve Walker, Dave Crocker, Dave Farber, John Seely Brown, Jeff Rothenberg. Mario Grignetti and Dave Crocker provided substantive comments on earlier drafts of this paper.

FOOTNOTES

[1]Recent mail is defined as mail which has arrived in a mailbox since the mailbox was last read.

[2]Cleaning up a message file consists of physically removing messages which have been marked as deleted from the file.

[3]The term user-model is used to mean the user´s mental model of the system. A mental model is a psychological isomorph of a physical system that represents an understanding of the system.

REFERENCES

[1] B. S. Borden, R. S. Gaines and N. Z. Shapiro.
The MH Message Handling System: User's Manual.
Technical Report R-2367-AF, The Rand Corp., Santa Monica, CA, November, 1979.

[2] J. D. Burchfiel, E. M. Leavitt, S. Shapiro, and T. R. Strollo.
TENEX User's Guide
3rd edition, Bolt Beranek and Newman Inc., Cambridge, MA, 1975.

[3] S. D. Crocker, J. Heafner, R. Metcalfe and J. Postel.
Function-oriented Protocols for the ARPA Computer Network.
In Spring Joint Computer Conference, pages 271-279. ACM, 1972.

[4] D. H. Crocker.
Framework and Functions of the 'MS' Personal Message System.
Technical Report R-2134-ARPA, The Rand Corp., 1977.

[5] D. H. Crocker, J. J. Vittal, K. T. Pogran, D. A. Henderson, Jr.
Standard for the Format of ARPA Network Text Messages.
RFC 733, The Rand Corporation, Bolt Beranek and Newman Inc. and MIT, November, 1977.

[6] D. H. Crocker, E. S. Szurkowski and D. J. Farber.
An Internetwork Memo Distribution Facility -- MMDF.
In Proceedings of the Sixth Data Communications Symposium. IEEE, Pacific Grove, CA, November, 1979.

[7] D. A. Henderson and T. H. Myer.
Issues in Message Technology.
In Proceedings of the Fifth ACM/IEEE Data Communications Symposium, September 27-29, 1977; Snowbird, UT, pages 6/1-9. IEEE, New York, 1977.

[8] T. H. Myer and C. D. Mooers.
HERMES User's Guide
Bolt Beranek and Newman Inc., Cambridge, MA, 1976.

[9] T. H. Myer and D. W. Dodds.
Notes on the Development of Message Technology.
In Proceedings of the Berkeley Workshop on Distributed Data Management and Computer Networks; May 25-6, 1976, pages LBL-5315:144-154. University of California, Lawrence Berkeley Lab., Berkeley, 1976.

[10] T. H. Myer and J. J. Vittal.
Message Technology in the ARPANET.
In NTC '77. IEEE, December, 1977.

[11] N. Neigus.
File Transfer Protocol.
In ARPANET Protocol Handbook, pages 265-298. Network Information Center, January, 1978.
Originally published as RFC 542, August 12, 1973.

[12] W. Plummer.
TENEX TECO: Text Editor and Corrector Manual
Bolt Beranek and Newman Inc., Cambridge, MA, 1973.

[13] D. M. Ritchie and K. Thompson.
The UNIX Time-sharing System.
Communications of the ACM 17(7):365-375, July, 1974.

[14] L. Roberts.
Computer Network Development to Achieve Resource Sharing.
In Spring Joint Computer Conference, pages 543-549. ACM, 1970.

[15] John F. Shoch.
Inter-Network Naming, Addressing, and Routing.
In Proceedings, COMPCON. IEEE Computer Society, Fall, 1978.

[16] J. J. Vittal.
MSG Manual
Bolt Beranek and Newman Inc., Cambridge, MA, 1976.

[17] M. D. Yonke.
BANANARD Reference Manual.
Technical Report ISI/TM-75-2, USC Information Sciences Institute, Marina del Rey, CA, January, 1975.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

AN EXPERIMENTAL MESSAGE COMPUTER SYSTEM BETWEEN UNIVERSITIES IN BRAZIL

Liane Margarida Rockenbach Tarouco

Data Processing Center
Federal University of "Rio Grande do Sul"
Porto Alegre, Rio Grande do Sul
Brazil

The long distances in a country with continental size restrict the scientific interchange between researchers. This paper descrives a project in order to improve the communication between universitary reasearchers integrants of the LARC (National Computer Networks Laboratory), by a computer message systems issued to give an environment able to support the computer conferencing activity.

1. Introduction

In October 1979, was created the LARC (Laboratório Nacional de Redes de Computadores) -National Computer Networks Laboratory. It is a Brazilian association between universities and research centers with the purpose of:

-To integrate the institutional efforts in the computer network area, producing nacional know how.
-To promote exchange of software and scientific information through the interconnection of local computation laboratories.
-To promote, through scientific exchange, the development of local capabilities that lead all associated institutions to a complete participation in the global efforts of the Laboratory.
-To make possible the development of interdisciplinaries researches of national extent according to local programs of the associated members of the Laboratory.
-To become model to others national programs in the computer networks area.
-To promote acquisition of experience in management of scientific and technological programs of national extent with a high level of sophistication.

One of the most important projects of the LARC is the implantation of a network interconnecting its associates. This network will have an experimental character, that is, to implement and test the most recent technologies of computer networks. This network, besides bringing about a commutation of messages between systems of data processing available in the associated institution (5) ought to support a computer conference activity. In this work it will be presented the way by wich we will try to implement such services.

2. Motivations

The idea of using the network to support a computer conference activity came up from two motivations. The first of them was to reach one of the main purposes of the LARC, which is to promote the exchange between Brazilian researchers. We must say that this exchange is specially hard because of the long distances between the Brazilian institutions, according to the map in figure 1.

Figure 1. Localization of the LARC participants.

The availability of a computer conference service will allow a more frequent conversation between researchers than that one gave by National Seminars, Symposiums and Congresses. We are trying with it to accelerate the development of the community by the equalizing of knowledge wich must allow to jump the gap by the groups more recently constituted.

This way of conversation, as described in (7), allow the participants to come to a discussion in any point without the risk of being lost. Distance and time sensations change themselves essentially with the use of such systems.

The second motivation to this project was a well succeded experience in the UFRGS(Federal University of Rio Grande do Sul). In 1978 it had been developed and implemented, though in experimental character, an application of computer conference based on the B-6700 system available in the University (UFRGS). This implementation gave a good experience in the management of messages between participants, including its storage in case of the receiver not being on-line, and also the possibility of doing a selective research in the historical file of messages.

We had realized how useful and interesting could be an analogous system to the purposes of the LARC, and then a contact was made with the researchers of the University of São Paulo who should make the hardware for the sub-network and the project was elaborated.

3. Objectives and purposes of the project.

The participants institutions of the LARC have equipments of computation received from different manufacturers. On the other hand, it is the Brazilian government orientation, the adoption of CCITT rules that necessarily bring us to the protocol established by the X.25 recommendation. The equipments installed in the network participant institutions do not have capacity to access it through the X.25 protocol. In such case, we established that the nodes of the sub-network, wich will form the substructure, where the messages will flow, have capability of dialogue with the external computers in these native protocol and between themselves, through the use of X.25. This gives a solution to the three first levels of protocol problem. The network will have the following schema:

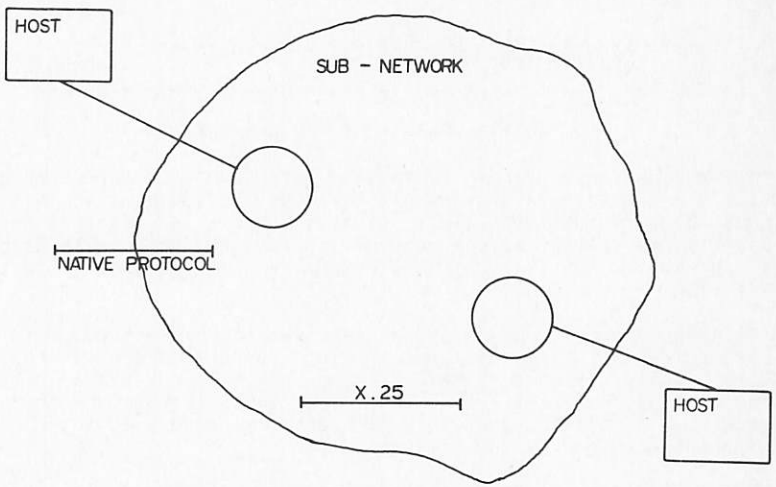


Figure 2. Schema of the network.

The nodes of the sub-network will have the incumbency of doing the protocol conversion. Each institution will be able to implement its own node, converting the X.25 to the most adequated protocol to its host system.

The aim of study in this work is the description of a protocol of higuer level to control the flux of messages between host, and to receive and answer the messages inherents to the computer conference activity. This will be done by the sub-network node. So each node of the sub-network will have capacity of external storage. The inteconnection between the nodes will be done using channels of the Brazilian network specialized in data transmission (TRANSDATA). For each node of the network it will be established one or more alternative routes so that will be assured the operation continuity, at least, concerning to data communication problems.

The studies for the development and implantation of the LARC network constitute another project and follow the models discussed in (2, 3, 4 and 6).

4. Description of the protocol.

The messages accepted by the system must have the minimum of format and the maximum of approach to the natural way of dialogue between persons. However, it is necessary to associate to each message a group of keywords in order to permit its recovery. The way chosen to signalize the keywords by the users was the annexation of a special symbol ("*") before or following a keyword.However it was observed that the user frequently used to forget to signalize the keyword(s) in his message. In the pilot experience, it was been necessary to give him an opportunity to signalize other words in the text, after the conclusion of its acceptance. Thus, just after the acceptance of a message, the system informs the user wich are the keywords associated to that message and accepts new indications of keywords. There must be at least one keyword associated to each message. The format of a message is the following:

```
I------------------ 80 characters --------------------I
I     PARTIAL TEXT                              . END I
I-----------------------------------------------------I
```

Figure 3: Format of the messages

A message must finish with the word "END". If more than eighty characters are used by a message, it will be presented in this size segments because this is the most usual size of a line in terminals. If the size of the segment of the message is limited, the allocation of space to buffering it in real memory and in secundary storage is simplified.

To inform the system who is the receiver of the message, the user can be guided by the dialogue or use a compacted form. In the first case, he digits the word MESSAGE or one of its sub-group with the minimum amount of letters enough for the instruction not being confounded with another one. This filosophy is being used in any others instructions accepted by the system.

The system then requests the code from the receiver. If the user does not inform anytihg because he does not know this code, the system accepts only the name and the institution addressing the message to an administrative department of that institution for conduction of the message. This will solve the problem of incorrect address as we use the help of a secretary to send messages incorrectly addressed. If it were necessary to accept only messages completely addressed it would be necessary to obtain directories too larges in each node containing, inclusive, receivers non users of the network. It is, however, possible to offer the services of an index by wich it can be solicited in any point of the network, the directory of catalogued users in any other node so that the address when possible could be complete and perfect.

In the case of regular users that do not want to go through every step of the dialogue, the system accepts instructions containing all the neccessary informations, separated by delimiters.

The messages, unless otherwise stated, are of free access, that means, they are available in an historical file to a posterior selective search. However it is possible to an user to associate two levels of security and privacy to his message. If he indicates, in the dialogue of acceptance of the message, that the message is a private one, the system will give him conditions to indicate two levels of treatment for the message.

In the first level, the user informs that his message can't be stored in the historical file. In this case, after deliver this message to the receiver, its content will be eliminated and only informations about statistic interest will remain on the historical file.

In other hands, it will also be accepted that the user classifies a message as private and able to be stored. In this case, only the emitter or the receiver of the message will have access to it, in the historical file.

In this protocol it has been avoided when possible conventions and rigid formats to the messages. The acceptance of a message from a dialogue conducted by the computer, facilitates the use of the system by a great series of users, inclusive laymans.

The messages waiting for the delivery to the receiver stay storaged in a system of management especially designed to this purpose, in whose node the receiver is associated. Each time the receiver does an operation to establish a session with the network, he will be informed that there are messages for him wich he can request. The delivered messages are put in a historical file where it will be possible to access them selectively by the emitter, receiver, dates and associated keywords.

5. The storage of the messages.

The software that will be implemented in each node of the network will have, besides the normal network functions, tasks related to the system of management of the messages. Some of the function to be implemented in the system are:

a. To receive and to store messages
b. To access, selectively, messages by receiver, emitter, date and content (keywords associated)
c. To catalogue the message doing the proper insertions in the directories
d. To modify and delete messages, under request or by expiration of time obeyng to established security criterions
e. To edit the messages for presentation
f. To compact and decompact the messages in the file to reduce space of storage

Other functions of the kind restart and recovery are:
h. To manage the storage area preventing overflow
i. To recovery or restore semi-destroyed files
j. To detect mistakes of address.

Users can send messages to another person (if he know the usercode) or to a office location (if he just know the local where the person works). In this case, the message is to be sent to a secretary in the office.

The system also will provide some line-oriented editing capabilities for correcting typing errors.

Users access the system using their name and a password. Once on-line, the user is told how many messages are waiting. He can then request a listing of the itens, including message numbers, senders'names, dates and subjects. He can read any of the messages, by identifying the message number.

The messages are filed in the node near the adressée. When participating on a computer conference, all conferents send their messages to the chairman. The chairman can send invitations to selected persons to participate on a computer conference to start it. A participant can send a particular message to selected conferee, during the computer conference, using the private message facility.

6. Conclusions.

The first experience realized in UFRGS with a centralized computer conference system (named MCS GRUPAL) gave some feedback that aid the elaboration of this new system.

The first feedback was to be know that the user only hardly can compose his message and also associate it the keywords, with simultaneity. So it was choose to make possible the two options before mentioned, to associate keywords to messages during its criation. Other data associated to the message for its posterior retrieval, like date, author, etc... are automatically supplied by the system.

When a computer conference is started, it receive a codified name that will be also a keyword associated to all its messages by the conferees. So it will be possible retrieve all texts belonging a specific computer conference.

Another feedback obtained on the first experience was to be know that it is necessary improve the message retrieval by a well structured file system oriented to minimize the searching time because answers must be presented in real-time to users.

This work, pioneer in Brazil, is justified either by the necessity to consider the purpose of the LARC through this application as well as this is an interestful area. Private enterprises will be able to use the result of this research. Probably, after its implementation the system will be adapted not only in inter university works but also in others environmemts. At first, the criterions of performance do not have priority. Only in the posterior versions and derivatives from experiences then obtained, the system will be reformulated to improve eventual failures detected on the performance. The pilot operation will also propiciate the collection of statistics and the weak points of the system refering to the recuperations from the failings in some of the components of the network.

This system constructed from a group of users requirements has the intention to serve them with a not expensive network based on microcomputers and storages units based on floppy disks being the nodes of transport sub-network interconnected by low velocity lines. Based on this it is expected to obtain a network useful to the national community and with a non-prohibitive cost.

REFERENCES

1.CHORAFAS, Dimitris. Data Communications for Distributed Information Systems. New York, Petrocelli Books, 1980.

2.DAVIES, D.W. & BARBER. Communication networks for computers. New York, Wiley, 1973.

3.TAROUCO,Liane. Redes de Comunicação de Dados. Rio de Janeiro, Livros Técnicos e Científicos, 1980, 2 edição.

4.SCHWARTZ,Mischa. Computer-communication Network Design and Analysis. , Englewood Cliffs, N.J. , Prentice-Hall, 1977.

5.TAROUCO,Liane. Laboratório Nacional de Redes de Computadores. In: SEMINÁRIO LATINO-AMERICANO DE COMUNICAÇÃO DE DADOS, Rio de Janeiro, SUCESU, 1979.

6.TAROUCO,Liane. Projeto de Redes. In: CONGRESSO REGIONAL DE COMPUTAÇÃO. , Belo Horizonte, Abril 1980, SUCESU.

7.VAHEE, J. Johansen & RATHLEEN, R. Spangler. The Computer Conference: An Altered State of Communication ?. The Futurist, 116-21, June 1975.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

MAIL2K : A COMPUTER MAIL PACKAGE FOR LOCAL APPLICATIONS

E. Motolese
E.D.P. Consultant
P.zza L. Da Vinci, 10
20133 Milan, Italy

R. Fondi, E. Salvadori, M. Vignali
CESAC S.p.A.
Via Ceresio, 12
22074 Lomazzo (Como), Italy

The paper introduces a computer mail package developed at Cesac S.p.A., Lomazzo-Como-Italy on a local network of UNIVAC 1100 computers. The package - which includes some basic message store & forward, computer conferencing, and telex interfacing facilities - was conceived as a logical extention of other more traditional services offered by the centre to its users toward a better support of all managerial functions. Its modular structure should ensure a balanced growth of the service in close response to user needs and environmental changes.

INTRODUCTION

Several studies in the past few years, have stressed that most of the savings offered by almost any kind of office automation are not as much related to the important changes this causes in the secreterial and administrative areas as to the small improvements it can carry out in the fulfilment of almost all managerial functions (this being an obvious consequence of the dramatic importance of managerial costs and time in all modern activities). [1,2,3,4,5]
In agreement with the above considerations the necessity was always felt at CESAC of providing its users with services, such as the general accounting package COGERT [6], which would account for the needs of the manager as well as for those of his administrative and secreterial staffs; and parallel to it the necessity of educating the managers to a proper use of the computer so as to take the maximum profit out of it.
Once the management converted to this active new role, it seemed obvious to think of new classes of services which would be explicitly addressed to this relatively small, though extremely important portion of the users community; and computer mail, i.e. the use of the computer to create, diffuse, receive and file information, immediately appeared as the most entitled to it since most of a manager's work is normally based on some kind of oral or written communication.
A more attentive study of the problem however evidenced the fact that this definite trend to a complete integration between information processing and trasmission (which is also at the basis of the well known phenomenon of convergence) was eventually going to determine a complication of the manager communication environment rather

than a simplification of the same, owing to the continuously growing number of alternative message services that were being independently developed and offered to him, such as TELEX, TELETEX, and all possible variations on the computer mail theme. [2,3]
The idea was hence conceived of a computer mail package, MAIL2K, which would include among its basic features the intrinsic ability to interface other systems, masking most of their peculiarities to the user and just leaving him some discretional power on the routing strategy to be followed when more than one could be chosen either in alternative or in combination (e.g. "go out on the TELEX networ immediately" or "stay on the private network as long as possible").
Furthermore, the above solution, which practically assigns to the service centre the important role of the communication node for all kinds of message services, also allows a uniform filing (and hence an easier retrieval) of all incoming and outgoing messages regardless of their origin, and through any one of the centre terminals to which a given user is entitled.
And last but not least, the devised solution provides the means to share among different users the cost of services which, otherwise, would not have been justified individually, and at the same time it ensures some extra traffic capability (and an adequate redundancy with resulting fault tolerance) to users requiring a full-time service on their own.

GENERAL DESIGN GUIDELINES

It is evident from the introduction above that MAIL2K was not conceived as a monolithic package including all possible features at one time.

Instead, it was clearly established from the beginning of the project that its structure had to be as modular as possible so as to allow a continuous uniform growth of the package in response to the more urgent requirements of its users, both for the introduction of new services and for the addition of new features to the existing ones.

Further, an open plan was defined for the growth of the package, which does state logical priorities in the realization of the single modules (e.g. the realization of mailing boxes logically precedes that of mailing lists) but leaves complete freedom in the solution of time conflicts as these arise (e.g. wheter the interfacing of the TELEX network is more or less urgent than computer conferencing).

Each new service appears in the plan as a new dimension in an n-dimensional space where single services may be upgraded alone (e.g. introduction of the mailing list feature in the TELEX service) or toward an integration with other services (e.g. extention of the basic mailing list feature to include users of different services). Consequently the plan must be periodically updated to include new services and features, not foreseen in the original plan, or to rearrange those priorities that were directly influenced by the latest

choices and developments (e.g. the early introduction of mixed mailing lists may significantly influence the modalities by which this feature can be extended to new services). The consistency of the revised plan is easily checked by means of a Petri-net model of the same.

The model can in fact immediately ascertain whether a given choice or development would actually dead-lock (i.e. make impossible) or live-lock (i.e. make improbable) other developments previously included in the plan.

The adoption of a dynamic structure as the one outlined above obviously requires that an adequate design methodology be adopted for the definition of the interfaces between the various modules.

For this reason, top-down design techniques were rigorously applied in the design of all MAIL2K modules, whereas a judicious use of data abstraction and nested virtual machines concepts and a definite recourse to interface opacity was used to avoid unexpected, and therefore more unpleasant, side effects in the event of maintenance of the modules. Furthermore, a recourse was made in this case too to Petri nets to check the consistency of the various interfaces and for the validation of all communication protocols.

Besides meeting the users' needs the moment these arise and choosing an adequate design methodology, several other factors eventually contribute to the failure or the success of a given product, among which it is worthwile to mention here the ease of use, the security and the cost of the offered service.

For that which concerns the first of these points, a transactional interface was chosen for the package because intrinsically easier to use, especially with the help of "menus", also by users not having a specific preparation in computer science or computer languages, as it can be expected in the case of the managers' community to which the package is explicity addressed.

Secondly, transactional interaction is intrinsically more reliable from the standpoint of the user in that it requires no special knowledge of the machine recovery mechanism in case of failure. Obviously it must be duly supported at the package level to avoid the loss of all established information and to alert the user of the exact amount of transient information that was preserved by the package trough the system failure.

Finally, for that which concerns the cost of the service, this is directly accounted on the user accounting file by means of a linear accounting formula including the cost of the transaction, the cost of the storage space for each outgoing message, the cost of external services if required (i.e. the TELEX charge), monthly charges for the input mailbox and secretary services, and possibly a monthly charge for being entitled to access certain external services.

The total charge of each mailing session and the partial total for the month is always displayed to the user at the end of the session for verification; furthermore, the accounting module can be expli-

citly interrogated by the user to obtain the cost of a service before this is used (e.g. the cost of a 400 characters TELEX to Tokyo or to Rome).

INTERNAL DESIGN

Besides of the general guidelines mentioned above, the internal design of MAIL2K was (and still is) deeply influenced by two major factors:
- the environment in which it was expected to operate,
- and the necessity that it be fully integrated with the other services/packages offered by the centre to the users.

Obviously the environment especially conditioned the choice of the languages used to write the different parts of the package and the list of o.s. features available to define the data-base and to handle the dialogue with the user.

More precisely, since MAIL2K was explicitly conceived to operate on a network of UNIVAC 1100 computers, the data-base software referred to was the UNIVAC DMS 1100, level 6R2 (but it will soon be upgraded to level 8 both to follow CODASYL recommandations as closely as possible, and to exploit its enhanced automatic recovery capabilities), and consequently most of the modules of the package are written in COBOL 1100, level 5R1 (X3.23 - 1974).

Similarly, the o.s. features used by the package are the standard ones provided by th UNIVAC EXEC8 o.s., level 36R2, and particularly its transactional interface package, TIP 1100.

Instead, the influence of the integration with other existing package and services was mainly concentrated on the various aspects of the user/machine dialogue and on the structural characteristics of MAIL2K messages.

In particular, the identification of the user was unified for all services and is achieved by means of the standard user-id/password mechanism provided by the UNIVAC 1100 Terminal Security System.

Once the user identified, a "user-id record" informs the system about all the essential data pertaining to that user, including the services to which this one is entitled and his access rights to the different sections of the data-base. After which the dialogue session develops under the continuous control of the machine and, for the unexperienced user, with the helpful guidance of detailed "menus".

Therefore, all MAIL2K functions can be invoked as any other service at any time of the session.

Furthermore, MAIL2K messages are physically subdivided into two distinct sections, i.e.:
- the header, which has an almost fixed format and contains the title of the message and all the mailing information;
- and the text, whose only requirement is that of being a legal EXEC8 file element;

thus allowing the user to directly handle as "texts" data records just extracted from the data-base, or results of previous computations, or program outputs, or even less usual items such as object files or plotting data.

The mailing process itself essentially consists in the creation of a chain of records which are used to link the message to the input mailbox of each addressee, to inform him of the existence of the message, and to the output mailbox of the sender, to allow him to monitor at any time the current status of the message (i.e. accepted, refused, expired, forwarded, etc.).

In the case of external addressees, i.e. addressers not belonging to the same node of the sender, the message is linked to a virtual addressee (namely a MAIL2K interface module) which, subsequently, re-arranges the message according to the appropriate line and network protocols, and eventually supervises its "mailing" until this is finally completed or aborted.

For complete transparency to the user all errors detected by the interface module are reported to the sender in the form of message refusals whilst the "short reply" feature is used to return the associated diagnostic information (e.g. the existence of illegal character sequences in TELEX texts).

MAIL2K messages are normally filed by maintaining the original separation between the header and the text, since the free-format of the record practically limits the possibilities of key searches to the data contained in the first.

The possibility is however given to the user to append a "tail" to the message to include all the keywords he thinks useful to add.

All phases of the MAIL2K and other services are protected trough the creation of temporary "session records", which are maintained troughout the entire dialogue session and are used to store all the information that can be needed to restore the dialogue to a consistent state after a system failure.

PRESENT STATUS AND SCHEDULE

At its first release, in early 1980, the package included only the very elementary features of a store & forward service [7], i.e. :
- a basic send function for the creation of outgoing mail;
- a basic receive function for the inspection of incoming mail;
- some text handling and text filing facilities;
- the accounting module.

Subsequently this same service was upgraded to include:
- mailing lists;
- a priority indication (ordinary or urgent);
- a restriction indication (open, personal, confidential);
- a validity indication (expiring date);
- a status indication (read, expired, refused, pending, etc.);

for the send function;
- automatic priority ordering of received messages (highest priori-

ty, nearest to expire, first issued);
- "copy to" and "short reply" (up to 80 characters) facilities;
- an automatic "re-route" facility (to be used in connection with the secretary service mentioned in the following)

for the receive function; and finally
- a transactional text editor

for further ease of all text handling operations.

Furthermore the following new services were added to the package in the same period, i.e.:
- a TELEX interface with automatic control of text validity (e.g. the absence of forbidden sequences), mailing lists, time-window indication, retry indication, automatic re-routing to up to 4 numbers;
- a secretary service for the automatic re-routing of all unclassified correspondence;
- an elementary computer conferencing service initially limited to users of the same local network.

At the present moment the following new features or services are at study or under development;
- mixed mailing list, e.g. inclusion in the same list of names of both MAIL2K and TELEX users (the feature implies the system ability to automatically verify the validity of the text for each user separately);
- message encryption (to be based on private and public passwords);
- partialization of message classification, i.e. messages containing both open and personal or confidential sections;
- directories of all possible destinators of MAIL2K messages (e.g. the TELEX directory);
- enhanced word processing facilities, e.g. for automatic text paging
- interfacing of IDN users (IDN standing for the new Italian Integrated Data Network);
- interfacing of at least one international computer network providing computer mail facilities (e.g. TELENET or TYMNET).

CONCLUSIONS

The MAIL2K package, a computer mail package developed for local application on a local network of UNIVAC 1100 computers, has been presented and its main characteristics grossly outlined.

Rather than on its features, which necessarily substantially recall those of most computer mail services, emphasis has been put on design methodologies and on the fact that the package has been conceived as a logical extention of other more traditional services as a help to the fulfilment of most managerial functions.

Furthermore the concept has been underlined that the service is intended to uniformly interface or integrate other services of the same nature rather that proposing itself as a possible alternative to them, thus assigning to the service centre the essential role of a communication node for all of its users.

REFERENCES

[1] POUZIN, "Recent Developments in Data Networks", Infotec State of the Art Report "Office Automation", 1980.

[2] M.A. SIRBU, "Communications and Message Handling in the Automated Office", Infotec State of the Art Report "Convergence: Computer, Communications and Office Automation", 1979.

[3] P.T. KIRSTEIN, "New Text and Message Services", IFIP '80, Tokyo, Japan, October 1980.

[4] H.L. MORGAN, "Research and Practice in Office Automation, IFIP '80, Tokyo, Japan, October 1980.

[5] E.E. MIER, "Tying together Telefones and Typewriters", Data Communications, April 1980.

[6] CESAC S.p.A., "Manuale d'uso del package di contabilità Integrata COGERT", Internal Document, January 1980, (in italian).

[7] D.E. SMITH, "Telemail User's Guide", GTE Telenet, Vienna, Virginia, May 1980.

CORPORATE MESSAGE SYSTEMS

Chairman - T. Myer

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP,1981

AN INTERNATIONAL MESSAGE SERVICE FOR CORPORATE USE

Ids Zandleven

Corporate Communications department
N.V. Philips' Gloeilampenfabrieken
Eindhoven
The Netherlands

The amount of message traffic within the Philips corporation justified already twelve years ago the introduction of a private store-and-forward message service. However, the growing needs and the new requirements demanded expansion of the network and the introduction of better services. These needs and requirements can be met by the concept of distributed message switching, the installation of more powerful and technical up to date systems, and the fruitful integration of the private message network and the private data (X.25) network.

INTRODUCTION

The N.V. Philips Gloeilampenfabrieken is a multinational company producing and selling a broad variety of electrical and electronic products and components. Its plants and offices are scattered all over the world thus requiring a well organized, economical and efficient use of the available communication resources. The public services, which are widely used within the organisation are telephony, telegraphy and mail.

TELEX AND TELEGRAPHY

During the past 15 years the public telex/telegraph medium has come of age. Within the Philips Company the advantages of telex have been recognized as an easy, efficient and economical method to transport messages all over the world, witness the tremendous growth in traffic volume.

Easy — because of the user-friendly operation which has been standardized all over the world.

Efficient — because of the fast and direct access to the required subscribers without the need of having the addressee available (unattended reception).

Economical — because of the low tariffs in comparison with telephone and cable or telegram.

Not only the advantages of the medium itself have caused the tremendous growth in the traffic processed daily by the Philips Communication Centre in Eindhoven (presently 17,000 messages daily of 750 characters on average), but also the post-war economic boom has contributed to it. Where Philips has to expand, communications has to follow.
This increase in telex traffic between Philips and a lot of other organisations all over the world was directly responsible in 1969 for the introduction of a private messaging network centrally controlled by a DS714/II.
The DS714 is a computer-controlled store-and-forward switching system, designed specifically for high efficiency control of message switching networks.

The fundamental requirement of this automatic message switching system is: extremely high system reliability, as the network is operational 24 hours a day, 7 days a week.

THE EXISTING MESSAGE SWITCHING NETWORK

The existing communications network comprises computerized message switching centres in Paris, London, New York and Hongkong, is interconnected via several private leased telegraph circuits and is connected to the public telex networks (see fig. 1). Nevertheless the DS714 system installed at the Communications Centre in Eindhoven still remains the dominating hub of the communications network.
Reasons for this centralized set up are:

- The centralized organisation (the Board of Management and the Main Industry Groups are centred in Eindhoven).
- Facilitate operations for non-Philips organisations to deal with only one centre and one telexnumber.
- Economical advantages can be obtained in gathering single messages for batch transmission on public circuits (untill recently a 3-minute minimum rate had to be paid for intercontinental calls).

By concentrating the intercontinental traffic in Eindhoven the volume of the message streams amongst the Philips organisations can be determined and controlled by the central message switching system. In doing so the company's transmission costs could be decreased significantly.

NEW REQUIREMENTS

As the DS714/II, after 10 years continuous operation, became technically obsolete due to the growing demands for higher transmission speeds and more sophisticated codes and features, it has been decided to replace the DS714/II by the newer, more up to date DS714/81. This system offers the special features for network requirements for a corporate data and message switching network. The DS714/81 has now been installed and has started its operational life.
In the meantime however, new requirements for the message switching network emerged:

- faster and more reliable transport of messages (error free, complete)
- message submission and delivery directly to and from the offices
- connection of data-oriented terminals, communicating word processors etc.
- extended characterset
- connection to computer mainframes

These requirements typically indicate a development towards smaller machines and distributed message switching.

The machine which fits in this development is the DSX40. It is a private automatic exchange for text and telex traffic. The DSX40 connects to a variety of asynchronous and synchronous circuits and accepts as its terminals normal teleprinters, data terminals and communicating wordprocessors.
The system allows communication with local terminals, with terminals connected to other message switching systems and with public telex subscribers.
Since the system performs local logging for statistical and accounting purposes it will be clear that the DSX40 offers us a path to a more economical and reliable distributed communications network and will bring the messages right into the office.

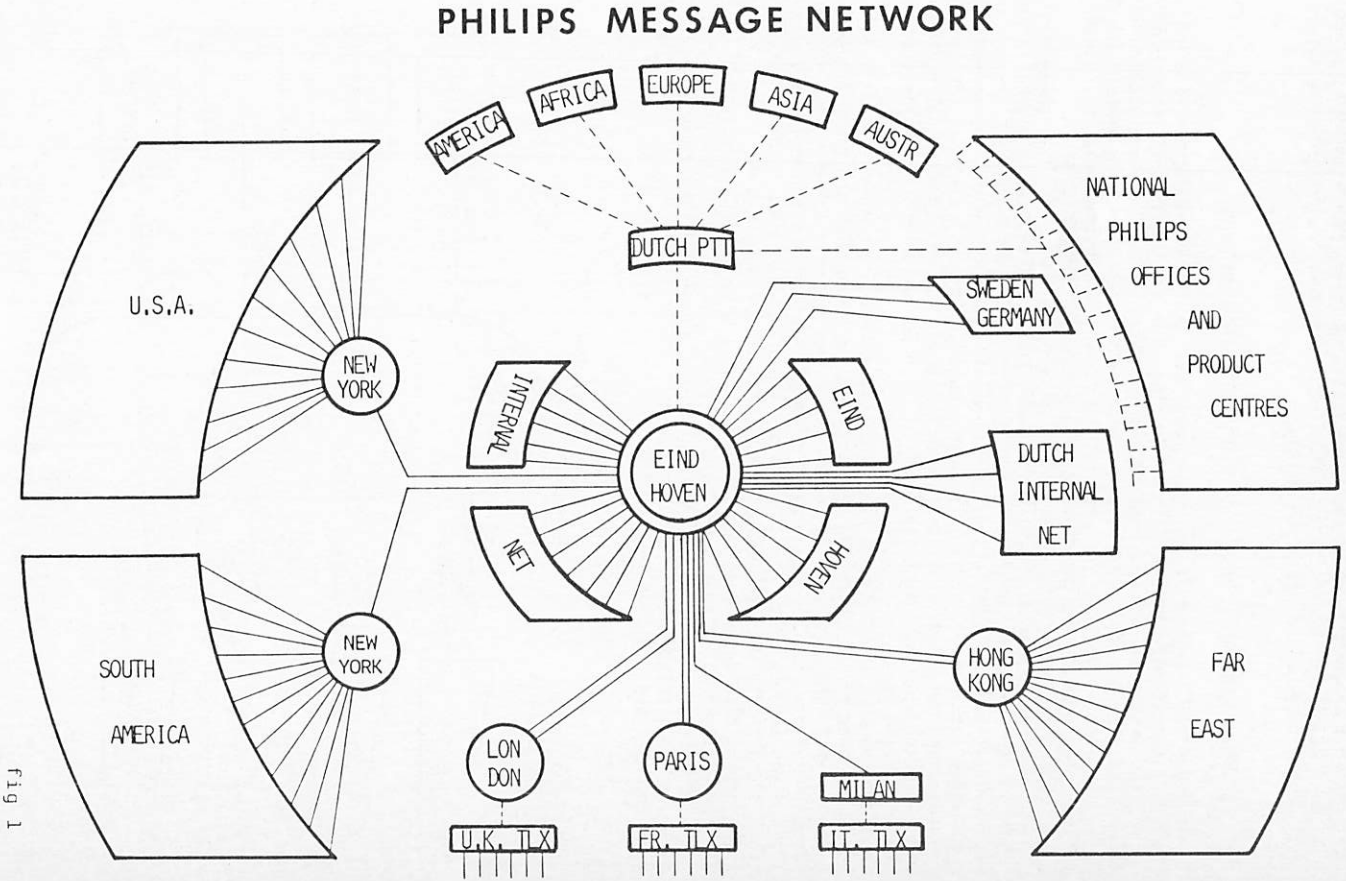

PHILIPS MESSAGE NETWORK
AMERICA
AFRICA
EUROPE
ASIA
AUSTR
DUTCH PTT
NATIONAL PHILIPS OFFICES AND PRODUCT CENTRES
U.S.A.
NEW YORK
SWEDEN
GERMANY
INTERNL
EIND
EIND HOVEN
DUTCH INTERNAL NET
NET
HOVEN
SOUTH AMERICA
NEW YORK
HONG KONG
FAR EAST
LON DON
PARIS
MILAN
U.K. TLX
FR. TLX
IT. TLX

fig 1

The development towards distributed message switching and the requirement for computer mainframe connections demand an efficient and reliable means of integrating message and data transmission.

THE INTRODUCTION OF CODAN, A CORPORATE DATA NETWORK

During the past decade it was not only the telex message traffic which went through a spectacular growth. Within Philips also (computer) data communications spread its influence, which of course brought growing pains and complicated ad hoc extensions. The growing demands for communication resources resulted in the last years of the seventies in an international network of leased lines and multiplexing modems: dedicated, complex and hard to control.
This comprehensible, but unacceptable development brought the management of Philips Corporate Communications to the decision to introduce a modern, advanced data communications network, called CODAN (COrporate DAta Network).
This private network consists of a number of switching nodes interconnected by high speed circuits. Each node is a minicomputer uniquely designed for efficient transport and switching of large streams of data. The network does not perform any processing on the user data, although it is able to convert data link procedures and character codes.
From the outside CODAN looks like a simple connect-through network. A user may connect his terminal to the network (dialled or permanent) and enter a username, password and host-destination code. If he is authorized, the user obtains a virtual circuit from his terminal through the network to the exit at which his destination has been connected. From that moment on it looks to him as if he were connected directly with the destination (a mainframe, a subsystem therein etc.).
CODAN offers also permanent virtual circuits and enables the connection of a broad variety of terminal-to-host procedures like TTY and BSC 2780/3780/3270, it offers an X.25 DTE/DCE interface and even a special interface has been developed to simulate plain telegraph circuits.
Some examples are depicted in fig. 2.

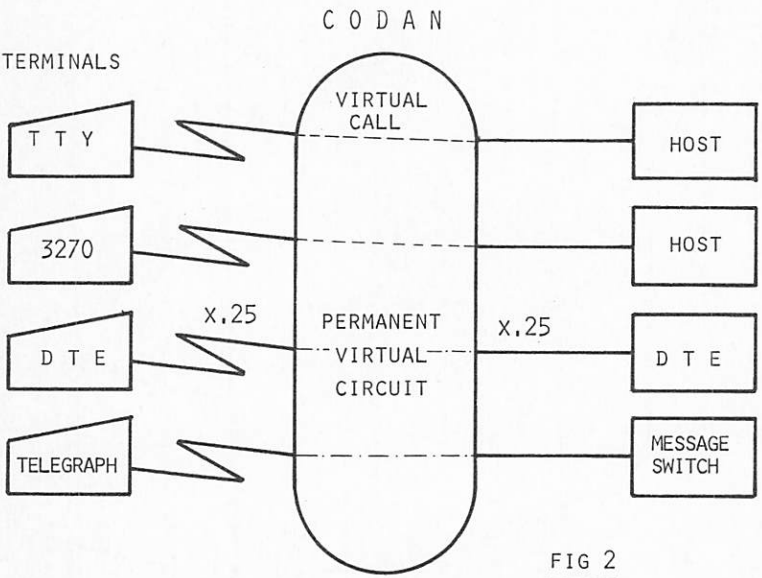


FIG 2

The geographical spread of CODAN covers at the moment six countries in Europe: the Netherlands, the United Kingdom, Belgium, France, Germany and Italy, as shown in fig 3. In the near future expansions are foreseen for other countries in Europe. Intercontinental connections are in study.

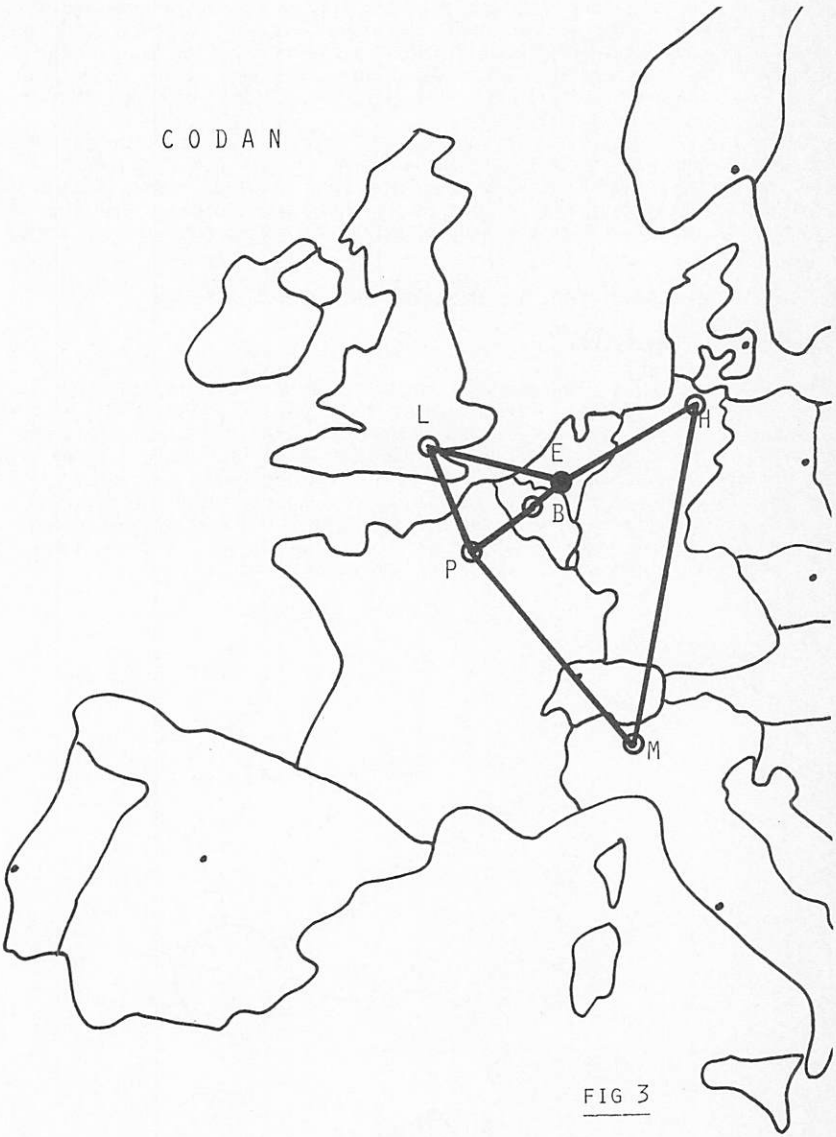


FIG 3

The most important capability of CODAN, when taking into account the needs for the construction of a distributed message switching network, is its ability to operate according to the CCITT Recommendation X.25. The importance will become clear when one observes the following interesting advantages of X.25:

- error control at link level which will significantly improve the reliability of the network.
- packet-interleaved multiplexing of the message streams. The packet level procedure of X.25 allows simultaneous transmission of several message streams, using parallel virtual calls and/or permanent virtual circuits. This logical multiplexing is obtained by packet interleaving at the single physical DTE/DCE interface to the network, which is highly efficient when one regards the difference in speed between this interface (e.g. 9600 bps) and that to the terminal (e.g. 50 baud or 300 bps).
- independence of character coding. This will make it possible to use IA2, IA5 and other alphabets virtually without influence on the packet transport.
- Its architecture fits in the Open Systems Interconnection architectural model of the ISO. Considering this fact as an advantage indicates our firm belief that the OSI model will be a standard for all future communicating systems.

INTEGRATION OF CODAN AND THE MESSAGE SWITCHING NETWORK

a. Geographical composition

The logical layout of the message network starts with the identification of areas where a relatively high flow of local message traffic is present or foreseen (e.g. geographical concentrations of plants and sales offices). Such an area can be served by one or more message switching systems, being a DS714 and/or one or more DSX40's, which form, when connected to each other, a local message switching network. A schematic example is shown in figure 4. The methods and procedures for interconnecting them is a local matter and may be a subject for second order optimization (shape and layout of the network, data link procedures to be used, protection and recovery etc.).

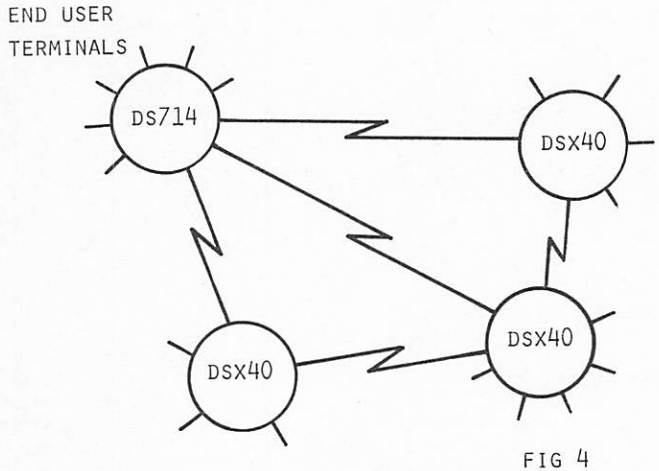
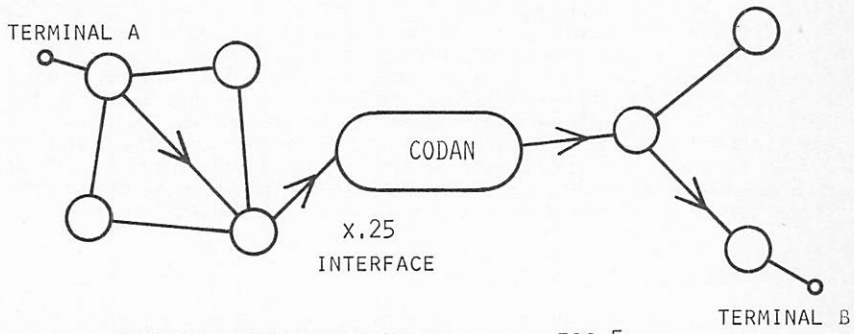


FIG 4

The interconnection of these local message networks will be achieved by using CODAN as an X.25 data network. Each local network will have at least one physical connection to CODAN. All inter-local messages will flow now via a switch connected to CODAN as an X.25 DTE as is shown schematically in fig 5.



MESSAGE FLOW FROM A TO B FIG 5

b. **Network architecture**

When comparing this complete **application** - network architecture with the already referred OSI architecture, we observe the following.
At layer 1 and 2 (physical layer and data link layer) there are no specific problems or abberations. Each physical and data link between message switching systems can and will be built by normally available techniques and procedures. Also at the DTE/DCE interface we meet a clean and comparable set of functions.
At layer 3, the network layer, the X.25 packet procedure fulfills indeed a great part of the functions to be performed here, but it provides no possibility to choose other networks or fall back circuits.
Looking at the transport and session layer we meet a still existing shortcoming of the OSI model. The exchange of messages in our store-and-foreward network is datagram-like and not session oriented. But still in the latest version of the model description 1 , which is a draft proposal now, we read on page 44: "Only connection-oriented services are defined in this document. Transaction oriented services and broadcast services are anticipated future extensions to the basic definition".
The consequence is obvious: we have to define our own rules and procedures in these layers. Particularly we point to the completeness check of the message streams, i.e. the requirement that no message is lost. Fortunately we have in the existing message network formats and procedures that cope with this problem, so we will at least copy them and improve them where possible with respect to technical and user-friendliness aspects.

c. **Addressing and routing**

Within the present operational network a private standardized addressing method is used for routing the messages to their respective destinations. The sender of a message must provide his message with a so-called routing indicator.
Multi-addressing, group-addressing and broadcasting are standard facilities.
The routing indicator is a 7-letter code of which the first two are a country code (adopted from the ISO standard 3166), the next two letters indicate an area and a switch in that area and finally the last three letters represent a

particular end-user in that area, connected to that switch. At present about 2500 different routing indicators are in use.
This 'hardwired' addressing scheme seems rather rigid; when some end-user moves to another area his routing indicator (= his message address) must be changed.
On the other hand, the benefits of this system are:
- a very fast routing of messages without extensive searching in long tables
- a visually recognizable relation with the addressee's place of residence
- it allows for e.g. area directed broadcasting.

Especially for the smaller sized message switching systems, which are introduced now in the network we find this routing mechanism to be very efficient. A local area network can easily detect whether the message should stay in the local area or should be sent to another one. In the latter case it has to be sent via the X.25 network and a table will indicate the DTE address of the destination area. The next step, the transmisson across the DTE/DCE interface is now an independent operation at a lower level.

X.25 SERVICES

One should realize that it is not enough to speak simply of X.25. The CCITT Recommendation defines several alternatives, facilities and options, and selection of the most suitable is not an obvious task. The 1980 version of the CCITT Recommendation X.25 2 offers principally three packet level services:
- Datagram
- Permanent Virtual Circuit
- Virtual Call

Datagram
This service deals with single packets only, which are in general too small to contain a complete message. Delivery confirmation can be asked for as a facility. Conservation of the packet sequence cannot be guaranteed, so software on a higher (end-to-end) level has to be used for control and restauration of the sequence of the packets. Such a sequence number should be added to the message data (per packet) and hence increase the overhead.

Permanent Virtual Circuit
This service operates with so-called packet sequences. A packet sequence is a sequence of packets chained together by the More Data Bit. Although not necessary, it is rather obvious to map a message onto a packet sequence. The sequence remains integral so no check on permutation of the sequence has to be performed. Delivery confirmation is an essential part of this service.

Virtual Call
Considering the transmission of data this service is the same as PVC. Extra overhead is introduced by the need for call set up and call clear packets.

Referring to the choice between these services we are at the moment still in the design phase. Some computation on the performance of the services, especially with respect to character overhead, makes of course the PVC favourite. For a final selection all variables of the services should be known such as throughput capacity, memory use etc.

NEW SERVICES

In the near future new telecommunication services will emerge like Videotex, Telefax and Teletex. Particularly the Teletex service, which is also based on distributed intelligent machines, will be an essential partner in message switching. In about the same way as we made 10 years ago a private telegraph service for world wide company use, we now plan to develop a private message service based on the CCITT Teletex definitions, built around the CODAN X.25 network and using the same distributed network of message switches. At this moment the provisionally adopted recommendations of Teletex are not well suited for a worldwide coupling of public Teletex services, not to mention for coupling with a private service. But we hope to find in time the appropriate means and methods to cope with these problems and thus continue this important communication service within the Philips company.

ACKNOWLEDGEMENT

The author would like to express his appreciation to his collegues for their help and comments and especially to Ben Sarton for his invaluable assistance in the drafting of this paper.

REFERENCES

1. I.S.O. Draft Proposal DP 7498: Data Processing - Open Systems Interconnection - Basic Reference Model.
2. C.C.I.T.T. VIIth Plenary Assembly - document no. 6, part III.2 : Recommendation X.25 .

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

THE DESIGN AND SERVICE IMPACT OF COCOS, AN ELECTRONIC OFFICE SYSTEM

N.W. Dawes S.J. Harris M.I. Magoon S.J. Maveety D.J. Petty

Bell-Northern Research,
Department 1R11,
P.O. Box 3511, Station C,
Ottawa, Ontario, Canada, K1Y 4H7

This paper describes the electronic office system COCOS (the Corporate Communication System) developed at Bell-Northern Research from 1978 to 1980. The design and impact of introducing this system into the company's offices is discussed.

INTRODUCTION

Bell-Northern Research's (BNR) computer centre is responsible for the provision of computer services to BNR. Early in 1978 this department realised that some form of electronic mail service could substantially improve BNR's effectiveness and efficiency. Numerous discussions and liaison with the potential customer divisions led to the setting up of a development group. This group was to develop and bring into service an electronic mail system integrated with word-processing functions. This paper describes how this system was designed and built, why various design decisions were made and how this has resulted in an effective, increasingly popular and mature service. The following paragraphs summarise its major user features.

Electronic mail can be narrowly defined as the submission, transmission and delivery of messages by electronic means. COCOS supports all aspects of message preparation, filing, retrieval, submission, delivery and receipt - as each user wishes. Messages can be memos, letters, reports or longer documents, addressed to as many or few people as desired. COCOS allows brief, normal forms of addressing and lets messages be similarly addressed to people, whether or not they are users of the system.

Work processing is a service which aids in the preparation of memos, letters, reports, documents and other textual materials. COCOS provides functions for word processing, and also allows the interconnection and use of communicating word processors.

Computer conferencing is a way of holding a conference without people attending at the same time or in the same place. Instead, the members of the conference, at different times and places, use computer terminals connected to COCOS. They can then type in contributions to conferences, read contributions and find out how up to date other members are. Since COCOS stores the contributions as they are submitted, members need only use a terminal occasionally.

COCOS is run as a production service, with high standards being maintained for availability, reliability, integrity and security.

SYSTEM DESIGN PRINCIPLES

There were several constraints laid on the design of the system. The service had

to run both on a PDP 11 computer, dedicated to the service, and on an IBM 370 computer, timeshared with other functions. The PDP version was generally intended for non-technical users. The IBM version was intended for the more technically orientated users of that system. The service had to run with these systems connected so that users on one system could send mail to users on the other without being aware of what system any user was located on. The connection logic had to permit extensive network growth. The IBM 370 runs the VM/CMS (1) operating system, but an operating system for the PDP11 had to be chosen. The system had to support electronic mail, document transfer, editing, formatting, and filing. Since the service had to evolve, it had to be extendable and modular in its design. The service had to be efficient in its use of computer resources, since it had to pay its way from charges for use. The system had to be operational within five months of setting up the project team (of five people). The user interface had to be easy to learn, provide appropriate feedback to the users and remain both simple and powerful. The interface also had to be so similar in the two versions that the same user manual could be used for both. The design of the two systems was recognised to be radically different, so each was attacked in turn.

UNIX (2) was chosen for the PDP operating system, mainly because of the excellence of its development tools and its adequate performance and maintainability record. UNIX interprocess communications are mainly accomplished by file transfer, and for efficiency had to be kept to a minimum. Large modules execute the major user applications (mail, editing and so on), rather than many small ones. This keeps down process communications and allows slightly simpler program structures. The switching between modules is done so that the user is only aware of a change in command environment (eg: from mail to editing). The syntax analyser has a minimum of links to the rest of the program. This allowed changes to the user interface and functions to be made without interfering with each other. The distribution of mail is performed by a single process on each node, asynchronously, external to the users and operating on queues of messages in files.

On the IBM system the service had to share the system with many other services. Therefore interprocess communications could not be via the file system so we used an IBM package (VMCF (3)) with extra networking interfaces developed by BNR. The major functions are executed by single modules structured so that syntax analysis is segregated. The routing and communications processes run externally and asynchronously to the users, operating on queues of files.

The message handling design is common to both implementations. However, no code or other design detail is common to them. This is because of language, operating system and file system differences. It would have been too timeconsuming and too machine inefficient to produce a single portable version.

MESSAGE HANDLING DESIGN

Message structure

The message structure was designed to balance several opposing aims. It has to be simple and governed by simple rules, as otherwise we would suffer from complex software problems and long implementation phases. The structure had to be able to accomodate all the data we might need to add for routing or addressing extensions. Such further accomodation could not occur at the expense of changing existing messages, but would have to be upwards compatible. The structure had to be capable of efficient data coding and decoding, so that it would not make the service too expensive to be useful. Finally, the message structure had to be easily readable by humans, so that debugging and maintenance could be done easily and quickly.

We chose a structure of two parts. The first part is the envelope and carries message data, addressee's names, subject and related data. The second part is the

body of the message, usually being the text of the message. The envelope is built in fixed length records, while the body is free form. A length of 80 bytes was chosen for all records. This is slightly longer than necessary to hold the information in the longest record used at present and is easily malleable by machines with different word lengths and file systems. Each record starts with a single alphanumeric character defining the record type. The structure within a record is fixed for each record type. In this way we allow for many record types and for record extensions as may prove necessary.

We chose fixed rather than variable length records for several reasons. Slightly more disk space and I/O is needed for fixed length records, but significantly less CPU is needed. The software for coding and decoding is much simpler and repair of messages in files is easy. The result was that the system became reliable much faster than if we had used variable length records. System performance is probably not significantly affected.

Message addressing

Two design decisions allowed very flexible addressing methods. First we designed message delivery logic which would deliver messages to people whether or not they use or are known to the service. Secondly we let users identify each addressee by the minimum information needed to deliver to that addressee. To help this we have a file with the names and details of all BNR employees and all people registered as using the system. This file is known as the corporate directory. The system then uses the name information given by the user to search this file to find further details about this addressee. These details include the department, location, and, if registered on the service, which node he is registered on and with what userid. If he is not registered as a user the user will be registered as being located at his local mailroom.

Users provide some or all of the following information in addressing correspondents: firstname or initials, lastname or names, department, location, organisation name. The minimum information needed is the lastname. If the organisation is not defined, the system assumes it is the sender's. Two main cases occur from all variations of information the user can provide. In the first case the addressee is known to the system (i.e. is in the corporate directory), in the second case he is not. In the first case the sender need only provide an unambiguous definition of this correspondent and the system will complete the other fields. If the definition is ambiguous the user is prompted to provide data to resolve the problem. The completed fields replace the original input. In the second case the system does not know about the addressee and will accept what the user gives. However, if all possible fields have not been completed, the user will be asked to give more information. He can then add information, change the original information, abandon composition or force acceptance of this addressee. Once the system has taken the address (during message composition) this address information is not altered by the system.

Users can present addresses to the system either directly or through a file called a receiver list. Such a list is a list of addresses in a file, one addressee per line in the file. The receiver list is invoked during composition of an address simply by inserting the name of the receiver list at that point. Receiver lists and normal addressee can be freely mixed in message composition or alteration.

Message delivery

Message delivery (routing) is the function of determining where a message shall be placed in order to get one copy to each addressee. On each node a single process is responsible for this function. This process handles one message at a time, and always processes a message to completion before starting on the next. Its queue of messages waiting to be processed is held in a file or files on disk. Processes

which are sending messages (ie: submitting them for distribution) can simultaneously append messages to this file or files.

When a message is sent a special file called a bundle is passed to the distribution process. The bundle header consists of fixed length records, like the envelope, and has records which define everyone who will receive copies of the message. The rest of the bundle consists of the envelope and body sections of the message. Bundles only exist during delivery processing.

Bundle header data is used in making routing decisions. A routing process only examines addresses in the bundle which have not yet received their copy. To assist this process, each node has a table which defines the communications exit from this node which will lead a set of nodes.

If the addressee's node name is that on which this router process is running in, the router has two choices. If the addressee's userid is provided, then a copy of the message will be delivered to the INBOX of that userid. If the userid is not provided the process will look up that addressee in the list of users registered on that node. If found there, the message will be put in that user's INBOX as before. Otherwise the message will be routed to the mailroom serving that addressee's location and organisation. Should no such location be served, the message will be sent to the organisation's default mailroom. When a message for a user is put in his INBOX, his name is crossed off the list of pending addressees in that message's bundle header section.

If the addressee's node is not the one in which this router process is running, the router has two choices. The choice depends on whether this process has already passed a copy of this message to any node. If he has not yet done so, he passes a copy to the appropriate communication process for transfer towards the destination node. If he has already passed a copy to any node, he does not pass a copy to any other. The only exception to this rule is when a destination node can only be reached from this node, in which case a copy is passed to it. The rule prevents the delivery of duplicate copies of a message.

Should no node name be present in an address, the delivery process can still determine where to pass the message. It has a table which relates one node name to each organisation. This defines a node in the network which can properly handle messages for this organisation. This routing process simply uses this node name as if it had been placed in the address it is presently checking.

This scheme allows setting up networks which can be wholly or partially interconnected. It ensures one copy, and only one copy, of a message is distributed to each addressee. Should addressing data be wrong, the system passes the message to the organisation's regular mail service to take care of it. It allows users to address across organisations without needing to know the exact details of the recipients names or registration details. This means people still can get away with making mistakes as they can with addressing conventional letters. Finally, users remain unaware of the network topology.

In practice this scheme has worked well, and the routing functions need little support. In a larger network, beyond 8 or 10 nodes, care would need to be exercised in designing the optimum communications paths for most rapid delivery of messages, as the cumulative effect of multi-node transit delays could otherwise become significant. Extension of this logic to handle much larger networks would require the addition of a further table to each node. In such networks there could be a multitude of organisations and many private citizens. To deliver to these one would need geographic information (such as state and country). This information would be used to find, from the table, a node which would have more information about this address.

USER FEATURES

Electronic mail

The main features of the electronic mail service are that users can create, send, receive and file messages using any terminal connected into a COCOS system computer. In the computer they are provided with a computerised INBOX and OUTBOX. Messages for them are put in their INBOX, and messages they compose and send to other people are generally held in their OUTBOX. Users can organise their messages in further boxes with names of their own choice. The computer is responsible for sorting and distributing messages according to whom they are addressed. Messages sent to people not registered on the service are printed out at a local mailroom and then delivered by the regular mail service.

Users can read, write, reply to, file, throw away (and retrieve again), messages as they would ordinary mail - except they now do it at a computer terminal. If there is a need to work with printed paper, users can have messages printed out and delivered to them by the regular mail service. The system keeps cross-reference indexes about mail kept in the computer, and it is easy to make the computer use these to help look for pieces of mail. The users can scan their mail for messages 'from' someone, or about a 'subject', or in other ways. Each message is uniquely defined by the name of the box it is held in (users can have many boxes, each with a separate name), and by the sequence number it is filed under in that box. With the box name and message sequence number a user can locate any of his messages in the system. (Naturally users are not allowed access to anyone else's messages.) Appendix 1 contains more details on commands, selection methods and system defaults.

Along with the normal electronic mail features COCOS also provides some more sophisticated functions. Experienced users can create their own commands from combinations of the mail commands, operating system commands (UNIX or VM/CMS respectively) and command control functions. These capabilities (further described in Appendix 1 under Macros) have allowed advanced users to readily develop new functions, without affecting other users. As well as being useful to individuals, the feedback from their experience has improved the choice of new features developed for the general users.

Some of the most important features of this service are its natural feel to the users and the ease with which users can tailor it. COCOS has an extensive set of personal tailoring options, which allow for both individual preferences and for terminal variations. For example, this tailoring might be to suit their particular level of expertise, which editor they prefer to use or the line width of the terminal they are using (see Appendix 2). It is this flexibility which has made the electronic mail service so successful, as the system behaves as the user wants rather than trying to make all users behave the same way.

Word processing

COCOS provides two main functions in its word processing service. These functions, like all others, may be used from any terminal connected to a COCOS system computer. The first function, an editor, simplifies the typing in and correction of text. This speeds up the process of production of correct text enormously. The second function, a formatter, takes this text and rearranges it so that it appears neat and tidy. The combination of these two functions, editing and formatting, both increases the speed with which letters, reports and so on can be typed in and considerably improves their appearance.

There are two main varieties of editor offered, which serve two main classes of terminals. Slow terminals, such as paper output ones and Visual Display Units (VDUs) operated on slow links, are best served by a line editor. Such an editor

allows simple typing in of text, display of one or more lines of text, insertion or removal of material at any point, easy moving around and searching in the text, and changing one or more lines at a time. The editor also allows for the inclusion of text from existing documents or messages. Fast VDUS are better served by a full-screen editor. This has all the qualities of the line editor, but allows many lines to be seen and worked on at the same time.

On the IBM system COCOS has integrated the IBM EDGAR and IBM CMS editors. These support fullscreen (3270) and line editing functions respectively. On the UNIX system COCOS has integrated the various normal UNIX line editors, one fullscreen editor and a secretary orientated line editor. The latter, TED, was developed on UNIX as a copy of an earlier version running over the DEC IAS operating system(4). TED provides editing features appropriate for non-technical managers and secretaries, yet is powerful enough to support COCOS programming development. One example of TED's extra functions is that it lets users type in text without them having to worry about line ends. It puts in line breaks at word boundaries itself, which speeds up input typing considerably.

The formatter offered is SCRIPT (4). (On the IBM version this is simply IBM SCRIPT/VS, on UNIX it is the major subset of SCRIPT/VS). SCRIPT can be used without any special knowledge to produce a neat letter or report from untidy text. It also has many powerful features which will alter line lengths, start new pages, set up headings and much more. SCRIPT is powerful enough to operate some of the most advanced automatic typesetting equipment in the world (eg: APS-5).

The formatter takes the author's text and formats it. This is fine if the recipients have terminals of the same width and depth of page. Often this is not the case, and COCOS provides a further formatting function to get round this problem. A user can define to COCOS the dimensions and characteristics of his terminal, and then get COCOS to format the text of the message so that it appears neatly on his screen. This is done in realtime through virtual terminal handling controls, which also provide further features. This is described in more detail in Appendix 2.

Naturally all these features in word processing can be used with the electronic mail service, so that letters, reports and so on can be neatly produced, checked and distributed.

All these functions are easy to use by people just starting with COCOS, and who have little background in such services. On the other hand they can be used by experienced personnel to rapidly and efficiently produce the varied and complicated structures of long documents. In this way, a wide range of needs is satisfied with users needing only to know the minimum required to do what they want.

Computer conferencing

Computer conferencing on COCOS was introduced in a multi-nodal fashion, so that one conference could have members on many nodes. By the time it was introduced for trials COCOS messaging and word processing features were well proven and accepted. Users could already broadcast messages and replies, and could maintain and track records of messages, message confirmations and so on.

In this climate conferencing was introduced in several stages. First the user interface for conferencing was made very similar to that for messaging, and identical at all points where the functions being executed were the same. The messaging interface was slightly extended so that conference members who used COCOS for messaging would be alerted by COCOS when new conference entries were made. This would let members check conferences only when necessary. These decisions removed the need for extra user training at this point. Then several conferences with trial groups were set up, and after these basic trials several further conferences were set up for users on demand.

The reaction to this service was quite unexpected. Users commented favourably on the user interface, and most conferences started off briskly. However, activity after a very few contributions behaved quite unlike that for messaging. In messaging the total volume steadily grows, except in holiday seasons. In contrast each conference went in fits and starts, and then petered out. This was not simply due to user inertia, but rather to the fact that practically all the useful work could equally well have been achieved using the messaging functions. What this indicates is that the features needed to provide most of the utility of computer conferencing are present in the messaging functionality COCOS provides. This is of considerable importance as it means the users need only learn one set of functions.

System access

Users access the service from many different types of terminals, and over several different access paths. Users of the UNIX version use general Ascii terminals, of speeds from 110 to 1200 baud. These terminals can be either hardcopy or VDU's and a very wide range of terminal parameter variation is supported (see Appendix 2). IBM version users can use either these general Ascii terminals (again with speeds from 110 to 1200 baud), or local or remote 3270 type terminals. Access paths to the systems can be across leased lines, DDD or packet switched with ITI support. The latter permits Datapac, Telenet, Tymnet and international packet switched access. In this manner users can use the system quite readily from North America and Europe.

OPERATIONAL FEATURES

Availability

Availability is the percentage of scheduled time the system can be fully used by any of its users. Clearly the scheduled time needs to cover the normal working hours at the very minimum. We have found it extremely helpful to our users to our users to run the service for as much of the time as possible. On the UNIX system we have one hour downtime per day, in the early hours of the morning; a four hour maintenance window late one evening a week, and a two hour backup period early Saturday morning. Since April 1979 we have achieved an average availability of 99%, running a medium size and a large size PDP 11 configuration. On the IBM system the scheduled downtime is longer, due to longer maintenance windows every morning. Also other programs share the system's resources. The average system availability for the COCOS users on the IBM system is 98.8% over the same period.

Reliability

System reliability consists of several components. The user, for example, expects that when he sends a message it will arrive at the correct destination(s). This is a very important part of the 'feel' of the service, and we have not let it be compromised at all. Again, the user expects command functions to work the same way each time, and as described in the user manuals. No surprises should be encountered. Also the data the user is working on should not be lost or destroyed, so that work in progress is not disrupted.

Message delivery reliability is a function of software integrity and hardware stability. To protect against the latter a copy of every message sent is placed on a second disk drive, so that in case of loss of data on the disk, the messages may be recovered. The software integrity has been achieved by the usual means: careful modular design, clear, documented and agreed protocols, organised programming and debugging and very close attention to problem reporting, analysis and repair. Our experience is that messages are not lost and that the delivery system is extremely rugged.

The command function reliability is attained by several means. In design we segregated the syntax analysis from the applications code, and used a formal process of bug reporting, analysis and fixing. The latter controlled the steps by which one gradually removes residual problems. Coupling this with a design which allows the fixes to be made without affecting other parts of the system, the system stabilised very rapidly in the first two or three months of use.

Integrity

Total data integrity is achieved when all the user's data is always preserved intact and uncorrupted, regardless of system failures. We settled for a slightly lesser standard, which is considerably less expensive. The data the user is working on is periodically backed up to another disk drive (usually once a night). Once a week the data is put onto tapes. Also all messages sent and delivered are copied to another disk drive, during distribution. Thus the maximum loss a user can suffer is to lose data worked on but still unsent. Such a loss could happen only if the data on the disk drive were unrecoverable. If the system crashes during editing, or similar activities, the normal loss is only changed data not yet stored on disk. This is usually no worse than loss of a line or two of editing

Security

User's messages and files are protected both by their userid and by their password. The userid need not be related to the user's name and the password can be changed by the user as he wills. Both userid and password are from 1 to 8 alphanumeric characters long. Users cannot read or access another's files or messages without knowledge of both their userid and password. Further, should a user attempt to login and fail to provide a valid userid and matching password, after four successive failures the line is disconnected. The means computer controlled attempts at security breaches are effectively deterred by continual delays.

Both the COCOS IBM and UNIX systems behave in the same way, except in some minor details, in all the aspects mentioned above.

GROWTH OF THE SERVICE

The pilot COCOS system was started up on the UNIX system in mid-February 1978, and the IBM version started two weeks later. Six weeks later the systems were connected together so that messages could pass freely between them. Early users, provided with a namual and a minimum of training, were encouraged to experiment with the system and to complain to a central user ('Real Problems'). The first 50 users provided effective and invaluable feedback on problems, inadequacies and difficulties. They also provided many suggestions for improvements. All this feedback was filtered and reacted to at high speed. In the first two months of service the average elapsed time from feedback to completion of fix (but not release) was less than two days. Positive acknowledgment of all information was given back to the users, and this, coupled with the rapidly improving service, lead to increased user confidence. In turn this lead to increased use and a consequent increase in requests for registration. During mid 1979 the word processing features were added; the line editor TED being developed for the UNIX system. In late 1979 the computer conferencing trials began. In early 1980 another PDP node was added, and both the original PDP and IBM hardware upgraded. In late 1980 a second IBM node was added as well.

From its small beginnings the service grew steadily in 1979, and then explosively in 1980. In 1980 it has averaged 15-20% compound growth per month in messages delivered, even with only a 3-5% growth per month in numbers of users registered. By June 1980 it had reached over 20000 message deliveries a month, with 650 users spread across three production nodes, working from Canada, the USA, the UK and Europe. Most of these users are managers and secretaries, with technical

writers, library staff, administrative staff and computer programmers providing the rest. Also many of the senior executive staff, with their secretaries, are using the system.

CONCLUSIONS

COCOS has solved several communications problems. Electronic mail in this form has enabled people to interact quickly, at a distance, and at times which suit each person. The work processing has improved the speed and accuracy of interactions. The service has been easy to learn by beginners, and yet powerful enough to satisfy most sophisticated demands. It has proven useful and acceptable to a wide range of people with varied jobs and skills.

The major benefits to users have been in handling their communications when they want. This is in contrast to suffering from incomplete phone calls, or phone call interruptions. COCOS allows users to communicate asynchronously. Since the users span time zones from the UK to California, this greatly increases the time window in which they can communicate.

The key user features of the service which have contributed to its success are as follows. Its addressing modes are similar in style and power to those used in normal correspondence. Messages can be sent identically to registered and non-registered addressees alike. Its command syntax is consistent and powerful, yet is easy to learn and use. Users have not been pressed to join the service, rather they have joined only when they wanted to. Finally the service runs with solid dependability, so that users trust it.

These are five general factors which defined COCOS's success. It was initially built without assuming we knew in detail what the users needed. Once a trial started, feedback from the users let us find out what the users's needs were. The design and development was flexible enough to respond to the emerging needs with enough speed to maintain the flow of feedback. Control of design and development was maintained so that escalating software problems did not emerge. The quality of the software and the service support were maintained at a sufficiently high level that the users trusted the system. In simple terms, we built a system quite close to the users' needs. We then modified the system to more closely match these needs. The system is therefore popular and accepted.

REFERENCES

(1) IBM Virtual machine facility /370:
Command language guide for general users: S370-36

(2) Bell System Technical Journal: July-Aug. 1978:
Vol 57 no 6 part 2

(3) IBM Virtual machine facility /370:
System programmer's guide: S370-36

(4) Script on minicomputers: S.J. Maveety
Share 51, Aug. 20, 1978.

APPENDIX 1

MAIL COMMAND SYNTAX AND MACROS

In the syntax below a 'box' is the name of a box, which is from 1 to 8 alpha-numeric characters long. A 'msg' is a form of message selection and an 'item' is either a 'msg' or a file name.

ALTER	box msg	Alter existing TO, COPY and SUBJECT.
BULLETIN	box	Create Bulletin line, TO, SUBJECT and text: addressees not shown to recipients.
COMPOSE	box	Create TO, COPY, SUBJECT and text.
DELETE	box msg	Delete message(s) (can be retrieved)
EDIT	box msg	Correct text of message.
QUIT		End use of COCOS.
FILE	box msg IN box	File in second box, delete in first.
FORWARD	box msg	Create TO, COPY and SUBJECT: text is present message plus user comments.
HELP	options	Give helpful information.
LIST	item	List boxnames and messages numbers in all boxes, or in chosen box, or type file.
PRINT	box msg	Hardcopy mailroom delivery to this user.
READ	box msg	Displays message(s).
REGISTER	box msg	When sent, confirmation given when read.
REMOVE	box	Remove empty (tidied) box.
REPLY	box msg ONLY	TO, COPY and SUBJECT set from message, user creates text. If ONLY replies only to sender.
RETRIEVE	box msg	Restores deleted messages.
SCAN	box msg	Give one line of details on message(s) in box.
SEND	box msg	Post message(s) for delivery.
SET	option value	Alter or check interface options.
SYS		Puts user in UNIX or VM/CMS environment.
TIDY	box	Destroy deleted messages in box, renumber rest

Message Selections.

a single number	eg: 6, to choose message number 6 only
a series of numbers	eg: 6,8,12 to choose these message numbers

a range of numbers		eg: 6-11 to choose messages 6,7,8,9,10 and 11
series and ranges		eg: 6,8,11-13 to choose 6,8,11,12 and 13
ALL		all messages in box
CURRENT		current message only
DATE	when	all sent on given date
DELETED		all deleted messages
FROM	lastname	all sent from person with given lastname
REGISTER		all registered messages
SUBJECT	string	all messages where the given string matches anywhere in the subject line.
TO	lastname	all with this name in a TO or COPY address
UNREAD		all not yet read by user
UNSENT		all not yet sent

Defaults

The system generally defaults to the most recently used boxname and message sequence number, should the 'box' or 'message' data not be specified. Initially the INBOX is the default boxname with message number 1. These other rules apply as well. If READ is specified with no parameters, the next unread message in the INBOX is read. If SEND is specified with no parameters, the last message handled in the OUTBOX is sent. If SEND is specified with no boxname, OUTBOX is used as the boxname. If SCAN is specified with no message number, ALL is used.

Macros

Macros let users execute a series of commands by typing only one command line. A macro is a text file with one statement on each line. These statements form a primitive interpretative language, which can be used to modify and create system functions. The statements are interpreted and executed one at a time, starting from the first statement in the macro file. A macro is invoked by the user typing its filename and any parameters he wishes, when at the mail or editor prompt respectively. Users can operate their own private mail or editor macros. They can choose one of the system-wide macro sets to use as well.

Mail macros have the following general capabilities. (Editor macros support a subset of them). A user can call a macro with up to ten parameters. The statements in the macro can be mail commands labels, unconditional skips to labels or skips to labels conditional on mail command success or failure. Parameter substitution is also allowed. Since macros can include mail commands which directly execute lower level command files (execs on the IBM system, shell files on UNIX), they are very powerful.

APPENDIX 2

PROFILE OPTIONS

Users can freely set, and examine their settings, of these options.

the mailroom location where mail PRINTED by him will be output:

his sophistication level (novice, intermediate or advanced) which controls the verbosity of the system responses:

which editor will be used for textual composition:

who apparently is composing his messages; though messages sent in such a manner will show the real sender's name in a SENT BY line:

which time zone messages he creates and send will be dated in:

language choice (French or English), all system responses and Help files will be output in the chosen language:

system wide macro directory name:

system wide receiver list directory name:

the width and number of lines that the terminal can display:

if the system pauses after a full screen has been displayed: when the user types carriage return to continue:

if a form feed is to be output before a read is started and after a screen full is displayed:

if the system adds delays to each line (eg: for hard copy):

if messages are displayed with or without the envelope portion specifying the date, addresses and subject:

what header is displayed at the top of each message when reading: headers can be null, '# boxname', or a long version used in mailroom routing of hard copies:

if full names and addresses or just the last name is displaying in the envelope of a message:

whether messages should be displayed as-is or with each line filled to the current line width set. This produces a ragged-right type of composition. This concatenation will stop or 'break' when a blank or tab is found in the first position of a line, or a blank line is encountered, or two consecutive carriage returns are found.

MESSAGE SYSTEMS AND OFFICE AUTOMATION

Chairman - H. Schuchmann

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP,1981

THE ROLE OF COMPUTER MAIL IN OFFICE AUTOMATION

Einar Stefferud

President
Network Management Associates, Inc.
Huntington Beach, California
USA

James McHugh

Office Automation
Hughes Aircraft Company
Fullerton, California
USA

Based on a review of general office requirements and the meaning of automation, we develop an activity/tool matrix to provide a framework for relating computer based message systems to office needs. Consideration is also given to the analogy between regular Postal Services and Computer Mail.

INTRODUCTION

Office automation has been touted by many as a nebulous, impossible-to-define buzzword around which we can't hope to put sufficient conceptual markers to permit a realistic view of the future. In this paper, we hope to show how obvious and inevitable certain elements of office automation are and how our knowledge of these markers should impact current plans.

Similarly, electronic mail has been acclaimed as the solution to many problems, but careful reading of the literature leaves us wondering if we really know what electronic mail is, or should be. In this paper, we hope to sort out the questions and issues surrounding this confusion and show in what direction the future lies.

OFFICES AND HOW WE USE THEM

One perspective is to view the office as "A PLACE WHERE INDIVIDUALS INTERFACE TO THEIR WORK". In our offices we write our texts, edit and format them, and package them for transmission to others. We receive mail from others, store it for retrieval, answer it, or perhaps extract it for inclusion in other documents we are preparing. We also meet with other people to exchange information and to make decisions. Our current offices provide us with text origination devices, editing devices, filing devices, display devices, and communication devices in the form of paper, pencils, typewriters, tables, envelopes, in/out baskets, cabinets, shelves, blackboards, dictation equipment, telephones, etc.

In one example, office systems provide the ability to conveniently track and maintain visibility over the work load of an individual, a group, or an entire organization, both from a worker's own point of view, and from a managerial overview. The system should permit origination and modification of information files by workers to report progress or annotate records, and by management to establish assignments, review progress and alter existing plans. These systems should allow for information, once entered, to be reused without reentry.

In a more general example, office systems should allow a worker to receive correspondence and other information via information transfer processes, and to

meld this with other information retrieved from files to prepare memos and reports to be delivered to others via the same transfer processes. Information should flow smoothly and freely among individuals and offices!

Whatever office work may be, we find that it consumes an increasing portion of the total work force, approaching 50% and threatening to dominate other factors of economic production. Thus we find ourselves deeply interested in finding ways to introduce automation to provide more efficiency and effectiveness, and to improve both the quality of the office product and the human qualities in the office work place.

YOUR OFFICE IS DIFFERENT FROM OTHERS

Visualize all of the physical objects in your office. You can see your desk, chair, tables, walls, doors, lights, and (if you're lucky) windows. How about the things on top of the desk like a telephone, address directory, clock, writing instruments, in-box and out-box, personal memorabilia and a calendar? Inside the desk, do you have scotch tape, paper clips, a stapler, a paper punch and note pads? Where are your files and bookcase; perhaps you have notebooks divided by topic. Do you have a typewriter or a terminal?

Does the appearance of your office differ from your peers? Are some of them able to operate with a desk four layers deep with the most current crash project on top, or is everything neat and well-arranged? Is their address directory a pocket pamphlet or a mechanical finder on their desk? Do they maintain files in the same manner as you? What about offices in other departments or companies?

How about when you take work home or when you travel? When you pack your briefcase, do you include a stapler, paper clips and calendar ... ?

If you stop to think about it, your office and the way you work is different from your peers. Whatever office automation turns out to be, we believe strongly that it must accommodate the same variety of behaviors that are implied by the questions above.

MEANING OF OFFICE AUTOMATION

Automation in the office involves use of automatic or semi-automatic mechanical and electronic devices in doing work. Office work includes working with information in various media, working with devices that process information, and working with other people. In talking about office automation, we are primarily concerned with providing automation tools in the office to help people do their work and to help them produce more with less human effort.

In the literature, we find that office work is called by many names: knowledge work, information processing, communication, correspondence, paper work, data collection, data analysis, researching, writing, report preparation, word processing, management, planning, staffing, directing, conferring, etc. In spite of efforts to characterize office work with one single all encompassing phrase, it would seem to remain a rich collection of widely different activities.

From experience, from the literature, and from time management seminars, we have learned that time is the most precious resource and that its conservation is most important in seeking ways to improve productivity in the office. We should be looking for ways to eliminate time wasting activities, and to avoid duplication of effort. Among the obvious is the technique of "NEVER HANDLING ANY ITEM MORE THAN ONCE". From this we come to some early conclusions about the desired outcome of office automation efforts.

One major conclusion is that data in any form (textual, numeric, or graphic) once entered into an electronic data file by means of keystrokes or by any other method should never again need to be reentered. Any changes to entered data, or extracts from it, should be achieved with text editing tools and should never require retyping of unchanged data.

A second conclusion is that office automation should provide individuals and offices with communication facilities that permit written messages, including various sized documents, to be sent at a sender's discretion and delivered in machine processable form to other individuals or to groups. The recipients should have the ability to use received messages as data for their own purposes, and be able to treat them with all the flexibility they have in handling files that they originate.

WHAT IS MAIL?

Whether it's US Postal Service or internal company mail, few of us give much attention to the manner in which an envelope arrives on our desk, and we have not given much thought to the definition of the word "mail". In this paper we define mail as:

> "Third party delivery of posted envelopes (containing information) to mailboxes located at addresses inscribed on the posted envelopes".

When we mail something, the first task is to prepare the contents (whether it's a casual note, a formal letter or a larger document). The envelope is then addressed, stuffed, stamped (if needed) and placed in a postbox. Once we drop the envelope in a postbox, the responsibility for delivery shifts to the mail carrier and we can forget it. The mail carrier holds responsibility until it is placed in the mailbox of the addressee. After delivery, the recipient is responsible for polling the mailbox to take possession of mail.

A key aspect of mail is the convenience of using a third-party carrier between your postbox and the addressee's mailbox. There are a number of different services that are widely advertised as "electronic mail" but which do not involve any third party handling.

It is important at this point to carefully consider this business of "third party handling". Without it, senders must each individually gain access to receivers' private spaces in order to deliver information. But with it, senders can rely on the postal service to gain access to the delivery point.

Mail provides a fundamental and pervasive service for society. In fact, it provides a backbone facility which enables many activities. What would happen in your office if the mail stopped moving? Could you use the telephone, hand-delivery, TWX and couriers to adequately sustain your business? Certainly the impact would be dramatic and would involve numerous distractions from your primary work to solve information transfer problems.

Considering how pivotal mail service is, it seems obvious that any alternatives to traditional mail must provide all of the same functionality with equal or better performance in terms of convenience, timeliness, and quality of communications. Half the functionality, even with 1000 times the performance, will not be acceptable. For example, if you could communicate in 1/1000th of the elapsed time but then could not forward received mail without rekeyboarding it, you would lose most of the potential benefits.

WHAT IS COMPUTER MAIL?

The elements of computer mail are directly analogous to those of paper mail. These include: Postboxes, mailboxes, envelopes, contents, addresses, carriers, senders and receivers.

Computer mail is different from other forms of mail because it uses electronic rather than paper-based media, and because it involves delivery of envelopes of electronic data to computer files which serve as mailboxes. Delivery does not require printing on paper, though the recipient might arrange to print it after taking possession. Analysis of some specific cases provides important insights:

FACSIMILE TRANSMISSION involves terminal-to-terminal copying of a picture image. One aspect is that paper is the most often required output medium, although there is progress toward digital text input with paper image output. A more important aspect is the requirement for simultaneous involvement of operators at both sending and receiving terminals. By our definition, facsimile transmission is not mail. But, store-and-forward facsimile services can bring it into the fold.

DIRECT TWX/TELEX TRANSMISSION is much like facsimile in that it uses leased wire or direct dialed connections between terminals with printed paper output, but it handles only alphanumeric data. Point-to-point word processor communication works in much the same way, except that it uses a word processor in place of the terminal.

STORE-AND-FORWARD TWX/TELEX NETWORKS can provide buffering between sender and receiver with switched network services which allow submisssion of a text along with one or more addresses to specify delivery points. In this case, the TWX/TELEX terminal plays the role of a mailbox. Note that it resides at a location that is specified by an address in the message header.

COMPUTER MAIL involves intra-computer transfer of data files (envelopes) from one user to another through a privileged third party process which has "write access" to users' mailbox files. A sender must connect to the mail computer to compose a file and send it, and receivers must connect to the same computer to access their mailboxes. One key aspect is that the full power of the computer may be used to process sent and received mail because mail is delivered in the form of data files. It may be printed on paper if desired, or it may not.

COMPUTER NETWORK MAIL involves inter-computer transfer of data files (envelopes) from one user to another through privileged third party processes which have "write access" to users' mailbox files. The network greatly expand the number of addressable users and essentially interconnects a collection of local computer mail systems.

COMPUTER CONFERENCING has been accomplished with computer mail systems by adding interaction procedures, control structures, shared mail files, and processes for analysis of conference proceedings. Conferencing is thus an application of computer mail and can be accomplished with single computers or a network of computers.

COMPUTER MAIL IN THE AUTOMATED OFFICE

Computer mail will play a major role in the automated office by providing primary transfer facilities for moving information in support of the full range of office activities. The following table shows a matrix of activities and tools among which mail will serve its purpose.

The Activity/Tool Matrix provides a framework for discussion of office automation in terms of Activities and Tools. For example, we can examine the differences between Personal Office Filing and Retrieval Tool requirements for Discussion and Correspondence versus Meetings and Conferences. The first needs individual privacy, while the second needs group shared access --- with some individual assigned to the filing chores. In the Packaging and Delivery Tool category, Discussion and Correspondence might favor full text copy delivery, while Meetings and Conferences might favor central filing with remote access by conference participants.

In another example, an analyst may draw information from institutional data banks, combine it with facts gathered from laboratory experiments, and distribute parts of a report to colleagues for comment by return message. Some returned comments might be incorporated directly into the report without retyping, and the final report might be entered into the institutional data banks as a published document. In this example we see that there are interactions among any and all categories, and that each interaction has its own character.

THE ACTIVITY/TOOL MATRIX

ACTIVITIES: / TOOLS:	A. Discussion & Correspondence	B. Meetings & Conferences	C. Fact Gathering & Analysis
1. Origination & Modification			
2. Mail- Packaging and Delivery			
3. Personal Office Filing & Retrieval			
4. Institutional Data Banking			

Activity Categories (Columns A through C)

Column A. Discussion & Correspondence covers exchange of large and small documents and messages among individuals and groups without declaration of a meeting or conference. Both formal and informal interactions are included. The work often involves individual one-on-one interactions, though other people are often copied.

Column B. Meetings & Conferences consist of recognized group efforts to collectively deliberate on some subject or to come to a conclusion and make a decision on some specific issue or issues. Both formal and informal group efforts are included.

Column C. Fact Gathering and Analysis focuses on collection and analysis of information to produce reports or documents. It includes source data collection, compilation, analysis, and filing, and includes accessing data banks to get information such as stock inventories, order backlogs, maintenance records, engineering data, personnel data, etc. It also includes accessing narrative text sources. Much of our past effort has been applied in this activity category to automate transaction systems and to automate scientific and engineering data collection, analysis and modeling.

Tool Categories (Rows 1 through 4)

Row 1. Origination and Modification covers text entry and editing, assembly and modification, formatting and proofing. This includes processing of text from received messages and documents, and other sources such as institutional data banks. It also includes capture and reduction of scientific or engineering data.

Row 2. Packaging & Delivery (including "mail" capabilities) covers assembly of texts into envelopes, addressing, transmission, receiving, acknowledgement, rerouting, etc. It might also include authentication facilities.

Row 3. Office Filing & Retrieval consists of the normal office activity of handling messages, correspondence and documents -- received, sent and in process. It includes all that information that individuals keep in their private desk drawers and office files. Many of these items contain narrative text.

Row 4. Institutional Data Banking & Retrieval covers the collection and accessing of institutional information and data, as is done with many current management information systems, or with scientific and engineering information systems. This category is included in the matrix because automated office mail systems must facilitate interconnection with, and utilization of, existing information in traditional data banks.

To reinforce the importance of computer mail in the automated office, imagine the matrix above without its packaging and delivery tools. Or, at least without any third party delivery system. Imagine being saddled with always having to transport your information by hand, or via paper mail services.

CONCLUSIONS

We concluded previously that data in any form, once entered into an electronic file should never again need to be reentered; that automation should provide individuals and offices with facilities that permit information to be sent at a sender's discretion and delivered in machine processable form to other individuals or to groups; that any electronic alternatives to traditional mail must provide all of the same functionality with equal or better performance; and that recipients should have the ability to use received data for their own purposes, and be able to treat it with all the flexibility they have in handling files that they originate.

Computer systems will play a vital and pervasive role in automation of the office work place. When computers capture all our information it will be imperative that we have computer mail services for electronic transfer of envelopes of data. Here is a compact review of what we should expect:

PROLIFERATION OF ELECTRONIC COMPUTING DEVICES. When $5,000 office computers can have the power of yesterday's $100,000 minicomputer, it will be hard to argue against placing them in local office applications everywhere.

MIGRATION TOWARD THE USER. Shifting computer/communication cost tradeoffs will lead to general migration of computation power toward specialty centers and toward the end-user. Powerful terminals will become as cheap tomorrow as typewriters are today. When this price equivalence occurs, terminals should become more common than typewriters, with at least one in every office.

ROLE OF NEW APPLICATIONS. Every day, new applications will become affordable in the office because of decreasing prices and many new applications will use small computers at department and project levels. This will stimulate demands for interconnection and interfacing to other computers everywhere.

NEED FOR LARGE COMPUTERS. Large computers will shift to specialized roles of providing central file systems and specialized computational services for shared user access by management and staff. They will provide the "backbone" systems upon which the proliferation of small computers must depend for file backup, software support and maintenance.

NEED FOR OUTSIDE SERVICES. There is no way that any company or institution can or should expect to provide every computation service with internal systems. It will always be economic to buy some specialized services from outside.

COMPUTER NETWORKS FOR INFORMATION TRANSPORT. Given the certainties of proliferation and migration, we must interconnect our scattered systems, including large central computers, local shared systems, individual single-user computers, and outside services. Once information is captured on electronic media, it will be unreasonable to print it on paper for transport to other offices where it would be retyped to get it back into electronic devices. Without computer network mail to provide transport, proliferated electronic devices can only trap information and impede the proper flow of information.

NEED FOR COMPUTER NETWORKS. Exploitation of new office technology requires installation of computer-communication networks to provide electronic information transfer among individual and office systems, and to provide mobility for information among office computers. The need is for an electronic media continuum equivalent to that provided by paper-based media today. Information on paper is highly mobile, while information in electronic files is often immobilized, in spite of the illusion caused by electronic transmission speeds between file, memory and terminal.

THE ULTIMATE NEED. In simple direct terms, the need is for complete network mail interconnection for proliferating electronic computing devices in our workplaces.

RECOMMENDATIONS

Explore the functionalities of advanced state-of-the-art computer mail systems. Gain practical day-to-day experience with computer mail. Do not limit your experiences to marginal improvements over TWX/TELEX, or confuse point-to-point communicating word processors with full function mail. Mail is a backbone facility for the office, not just another feature or function key on your keyboard.

Evaluate new office automation tools on the basis of compatibility with computer messaging systems. If new tools will not facilitate interconnection with computer mail systems, then they threaten to immobilize your information, and thus rob you of the highest potential payoffs of office automation. Any computer based tool should be connectable to a computer mail service to allow sending and receiving.

ACKNOWLEDGEMENT

The ideas in this paper have been developed in the context of the reports, articles and papers listed here. The authors acknowledge the debt owed to the many individuals who have contributed to the the work represented here, though responsibility for the conclusions and statements (and errors) must rest with them alone.

REFERENCES

Bolt Beranek and Newman, Incorporated, Specification of a Draft Message Format Standard, National Bureau of Standards Report No. ICST/CBOS-80-2, Washington, D.C. (October 1980).

Crocker, D., Alternatives in Electronic Mail Networks, DATA COMMUNICATIONS Magazine, Vol. 9, No. 10, (October, 1980) p 95.

Crocker, D., Electronic Mail, Part II: Network Integration, DATA COMMUNICATIONS Magazine, Vol. 9, No. 11, (November 1980) p 109.

Crocker, D., Vittal, J., Pogran, K., and Henderson, D., Jr., Standard for the Format of ARPA Network Text Messages, Request for Comment (RFC733), in ARPANET PROTOCOL HANDBOOK, NIC 7104, Network Information Center, SRI International, Menlo Park, California, November (November 1977).

Gilbert, J., et al, (Directorate for Management Information Systems), DARCOM Blueprint for Information Processing in the 1980's, U.S. Army Materiel Development and Readiness Command, Alexandria, Virginia (October 1978).

Lukasik, S., Lichtwardt, D., Maxwell, E., McKinney, J., and Roberts, C., The Future of Electronic Information Handling at the FCC (Blueprint for the 1980's), Federal Communications Commission, Washington, D.C. (October 1980).

Pickens, J., Formal and Informal Message Services Development at SRI International, SRINet Technical Note SNTN No. 6, Menlo Park, California (August, 1980).

Raphael, B., Heathman, M., Rom, R., and Stefferud, E., Proposal for Development of SRI Network Facilities, Submitted by Computer Resources and Telecommunication Science Center to SRI International, Menlo park, California (February 1979).

Stefferud, E., Economic Background for Computer Mail: A Review of Fundamental Issues, in: IEEE Computer Society, DIGEST OF PAPERS: COMPCON 80 (Spring), February 25-28, 1980, pp 94-97.

Stefferud, E., Electronic Mail and Computer Mail Systems, The OFFICE Magazine, Vol. 91, No. 5, (May 1980), p 50.

Stefferud, E., Electronic vs Paper Media Continua -- A Comparison, in: National Computing Conference 1980, PERSONAL COMPUTING DIGEST, pp 25-28.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

RESEARCH ON THE IMPACT OF OFFICE INFORMATION COMMUNICATION SYSTEMS

DON TAPSCOTT
Bell-Northern Research Ltd.

Integrated office systems can improve the effectiveness of office workers. This paper describes research conducted at Bell-Northern Research, Ltd., Custom Systems Division, in Toronto where 19 knowledge workers were given electronic work stations to help them with various aspects of their jobs, including text editing, electronic mail, information retrieval, and numerous administrative functions. Some posttest and system monitoring findings are presented and a methodology for "user-driven design" or customization of a electronic office systems to a given office is discussed.

1.0 Introduction

When Will Rogers was informed that U.S. President Calvin Cooledge had died he replied: "How can you tell?"

Unbeknownst to Rogers, his amusing question has become one of the burning issues confronting the providers and potential users of Office of the Future technology.

That is "how can you tell" that the new electronic office systems actually do anything positive? How can we measure the impact of these systems on the productivity of office workers and organizations? What is the hard evidence that these systems can improve office efficiency and effectiveness?

Intuition and early experience indicate that integrated office systems can have a dramatically positive effect. The integration and application of computing, telecommunications and information technologies to the office environment is already beginning to transform the ways in which people and organizations work. Moreover, it appears that the greatest potential of these changes will not be cost savings from increased typing efficiency, but rather, the augmentation of the abilities and effectiveness of the "knowledge" worker. That is, the new electronic office systems hold the potential to transform the capabilities of those who work with their minds in processing, managing, retrieving and communicating information.

However, until recently there has been little work done to measure the impact of integrated office systems on the effectiveness of knowledge workers and organizations.

The lack of answers to the question "how can you tell" poses a serious problem for just about everyone touched by the new technologies. Product planners, system designers, market analysts, along with implementors, users and "choosers" in the office are all forced to work in a vacuum. Without valid measures and data quantifying what these systems do, it has been difficult to plan, design or market them. It has also been difficult for those considering using office systems to measure their needs, customize systems to their environments and measure system effects. As a result, many systems have been inappropriate and have failed. Others, lacking a clear and hard dollar cost-justification have never made it out of the lab.

Research underway at Bell-Northern Research in Toronto is attempting to grapple with this problem. A research project, funded by Bell Canada, was initiated in early 1979 to develop measurement techniques and collect data on the impact of advanced office systems. The work was conducted by a multidisciplinary team at BNR, with the help of other behavioural scientists at Bell Northern Research and BNR Inc.

Nineteen knowledge workers at BNR were given "electronic work stations" to help them with various aspects of their jobs. They used the system for

- ⊕ electronic mail,
- ⊕ information retrieval,
- ⊕ word processing,
- ⊕ administrative support, and
- ⊕ data processing.

These workers' attitudes, use of time, communications patterns, etc., were compared to a control group in a pretest-posttest, quasi-experimental research design. As well, the users activity on the system is being monitored on a day-to-day basis.

The results of the posttest, after 8 months of use, indicate a number of important improvements. Using these data it was possible to make a business case to extend the pilot system into a full operational office system for use by most employees in the company.

2.0 The Research Project

Office system research can be grouped into four categories:

- ⊕ laboratory experiments
- ⊕ true experiments in field (office) settings
- ⊕ quasi-experiments in field (office) settings and
- ⊕ non-experimental designs. (Tapscott, 1979)

Most research to date has been of a non-experimental nature, in particular The Treatment Group Posttest Only Design (usually called the Case Study). While such designs may be useful in exploratory research to aid in hypothesis formulation, they are usually not adequate to permit testing of hypotheses.

There have recently been, however, a number of laboratory experiments examining the impact of systems, in particular computer conferencing systems (Hiltz and Turoff, 1978,76). Such experiments have produced some very interesting and useful results. However, laboratory experiments have limitations in office system research at the present time. These include:

- ⊕ the limited scope of the laboratory to recreate important aspects of the office environment
- ⊕ their limited external validity (generalizability of the findings).
- ⊕ the current early stage of hypothesis formulation, making it difficult to test precise hypotheses.
- ⊕ problems obtaining representative samples of knowledge workers, and
- ⊕ the lengthy time involved to become fully proficient on a system.

The most appropriate designs at this stage of the research tend to fall in the category of Quasi-experimental Designs in Field Settings. These designs involve some experimental controls, but lack random assignment of subjects to groups.

The design of the BNR pilot research is a quasi-experimental design which is configured as follows:

O	X	O
O		O

O = Observation

X = Treatment

FIGURE 1
O.I.C.S. PILOT DESIGN

An experimental group (n=19) and a control group (n=26) were pretested (observation) in August 1979. The O.I.C. System (the experimental treatment) was then implemented immediately. The posttest was conducted in May, 1980.

Research using this design, if properly conducted, can eliminate many of the threats to internal and external validity and result in interpretable data. It may be possible to indicate that observed differences between experimental and control groups at the time of the

posttest are to some degree caused by the experimental treatment (system use). For example, it may be possible to eliminate the following threats to internal validity:

- ⊕ history. Posttest changes in effectiveness, for example, could be due to events occurring between the pre- and posttests in addition to the introduction and use of the system. However, such events should also affect the equivalent control group in a similar manner. If they do not, then it may be possible to attribute causality to the treatment.

- ⊕ selection. If observations between control and experimental groups were made only once, one could not determine if any observed differences were due to differences between treatments, or whether the two groups were initially different. However, with the use of a pretest, this possible explanation can be eliminated.

Several other features of the design are: There were 3 participant observers in the experimental group. These O.I.C.S. project members act as full members of the pilot, to directly observe the various facets of system use and to fully participate in learning the system and using it to help them with their jobs. Pre- and posttest measures are taken to ensure that participant observers do not differ significantly from other members of the experimental group.

1) There is ongoing monitoring of experimental group activities on the system. Measures built into the system include login hours, commands used, applications used, communication patterns, etc. In fact there is a record of all activity on the system. To insure privacy and resulting respondent trust and co-operation, the content of messages, text and files is not being examined at this time.

2) A pre-experimental survey of management was conducted before the pretest. This survey enabled the research team to determine management perceptions of what constitutes an improvement within the organization. This information facilitated both the design of the pretest instruments and the system itself.

3) Structured interviews were held midway between pre- and posttests. These interviews were used to help assess initial user progress in learning the system, assess needed system modifications, and finally continue the training process.

The measurement instruments developed for the BNR pilot include:

- ⊕ a prestudy Organizational Effectiveness questionnaire
- ⊕ an extensive survey questionnaire
- ⊕ a Network Interaction Analysis form asking respondents with whom they communicated in a

"typical" specified time period
- ⊕ an Activities/Communication log
- ⊕ a method of system monitoring, and
- ⊕ structured interviews.

The relationship of these instruments to the research design is indicated in Table 1 below.

TABLE 1
MEASUREMENT INSTRUMENTS

	PRETEST	POSTTEST
1.	Questionnaire I ⊕ system Rm requirements ⊕ communication data ⊕ information use data ⊕ attitudinal data ⊕ job design data ⊕ demographic data	Questionnaire I same, plus items on attitudes towards the pilot.
2.	Network/Interaction Analysis	Same
3.	Activities Communications Log	Same, plus data compared to #4 below Data generated from the system. Posttest interviews of experimental group.

2.1 The Sample

The sample consisted of an experimental group of 19 and a control group of 26 BNR employees. The experimental group consisted of 7 managers, 8 professionals, and 4 administrative staff. The control group consisted of 9 managers, 11 professionals and 6 technical staff.

As explained by Tapscott et. al. (1979), in office system research it is not feasible to randomly sample a given population and randomly allocate subjects to control and experimental groups. Consequently, pilot and experimental groups were selected using other criteria. Some of these had more to do with the internal needs of the company, rather than to enhance the quality of the research. Most, however, were related to both needs, as the two sets of needs are generally not contradictory but complimentary. That is, a good research design and system must correspond to genuine needs of the workers and organization involved to be successful.

A standard "Decision Analysis" was performed in which each of the criteria were given weights. Each alternative configuration was then scored as to how well it met each of the criteria. The products of these scores by the criteria weights were summed resulting in a total score for each of the alternative system configurations. The configuration with the highest score was then selected.

2.2 The O.I.C. System

The O.I.C.S. pilot system was developed, not as a product, but as a vechicle to enable the BNR research into the impact of office systems. The system consists of an integrated set of application programs running on the UNIX (registered trademark of Western Electric) operating system. Access is provided through terminals that are hard-wired CRTs without any special function keys or other intelligence.

A user logs on to the system by giving a unique user identification name and password. Once his or her identity has been established, the user has access to any part of the system environment through typing the command name associated with the facility desired.

For example, the user accesses the Bell and Northern corporate electronic mail system by typing "cocos", the automatic diary and To-do tool by typing "diary", an on-line bibilography by typing "library", the interactive facility to make a paper letter by typing "mkletter", and the basic calculator by typing "bc".

Recent enhancements to the system enable the user to select a user interface. For example, some users may prefer a menu-based interface. Here menus appear on the screen. When the user wants to do something on the system s/he simply types the number associated with the menu item desired. New users or users who run into trouble are assisted by a Computer Aided Instruction capability.

Some of the features of the system are:

1) Communications Tools

- ⊕ Electronic Mail. The "COCOS" system, developed by BNR enables users to compose, send, forward, reply to and file electronic messages.
- ⊕ Paper Correspondence. The user is prompted by the program which automatically generates formatted memos and letters.
- ⊕ Maillog. This system enables the tracking of paper mail in, out, and within the office.
- ⊕ Synchronous Messaging. Users can, if they permit it, interrupt each other on their screens with short urgent messages.

2) Text Processing

- ⊕ The text editors. Several are available. The most commonly used is a line oriented text editor with a terse user interface.

- ⊕ The formatter. A powerful text formatter called NROFF structures the input of the text, e.g., chapter headings, paragraphs and the like, and outputs reports with pagination, hyphenation, justification, page numbering, point numbering, table of contents, etc.
- ⊕ Spelling Checks using three dictionaries as databases
- ⊕ readability indices for examining the content of text
- ⊕ sorts, merges, text and data manipulations
- ⊕ some table, figure and graphics capabilities.

3) Information Retrieval

The system provides an information retrieval sub-system to maintain and query databases of any type of information. Among the first uses of this facility in the O.I.C.S. study were a project bibliography and a conference and seminar schedule.

Future uses will include direct linkage to numerous databases currently only available in the Company's Technical Information Centre as well as both customized and Company-wide databases for project data tracking, status reporting and milestones, etc.

4) Administrative Functions

Among the administrative applications currently employed by users are :

- ⊕ personal log (what was done when)
- ⊕ todo file
- ⊕ cost tracking
- ⊕ schedules
- ⊕ coming events
- ⊕ time reminder
- ⊕ phone lists and telephone area codes
- ⊕ acronyms
- ⊕ desk calculator.

5) Analytical Tools

- ⊕ Interactive Statistics. This package enables a variety of statistical applications with simple graphical output, ranging from calculation of means to regression analysis. The user is able to interact with the data on the screen.
- ⊕ Basic Calculator. This is used for basic arithmetic and scientific operations.
- ⊕ Data processing. Sort, merge, etc., facilities enable the manipulation and processing of reasonably large amounts of data.

2.3 Limitations of the Study

The BNR study applied more rigorous scientific methods to office system field research than other studies to date. In addition, tests of selected data conducted by researchers at the University of Waterloo indicated extraordinarily high data reliability. Despite these positive aspects, there are a number of important limitations to the study which are discussed in detail in Tapscott et. al. (1980). These limitations are related to:

1) difficulties performing field studies in general, and office system field studies in particular;

2) The uniqueness of the office environment where the study was conducted; and

3) organizational changes, staff turnover, and changes in the content of work in the office environment over the ten month period.

These problems have their greatest impact on the ability of the researchers to generalize the findings to broader populations of office workers. Consequently, while the findings of this study may well be the most valid from office system field research to date, caution must be applied when attempting to generalize these findings beyond the specific experimental environment. Rather, it is best that the results are seen as an investigation into the impact of one electronic office system in a given environment. As such they serve to present more detailed and solid hypotheses for verification in other situations.

3.0 What We Found Out

Many of the results of the study are not yet available for release by Bell Canada. However, it is possible at this time to describe some of the findings after about a year of system-use. A number of the original hypotheses regarding how the office system would improve efficiency and effectiveness were confirmed. Some hypotheses were not confirmed, and in some cases new hypotheses were generated. In general, it appears that the system had a positive impact -- improving

communications, time-use, access to information, attitudes towards office system technology, and quality of working life.

1) System-use

The system accounting capabilities give a picture of the extent and character of system-use. The mean login time per day (January - June) for the members of the Pilot group who were part of the group from the inception of the study (Original Pilot group) was approximately 3.7 hours. A second statistic considering both the Original Pilot group plus all people added to this group due to attrition (Ongoing Pilot), indicated that mean login time was approximately 3.1 hours per day.

Use of the computer messaging facility has steadily increased. During the first nine weeks the average number of messages sent per day per user was only 2. During the last nine weeks this figure had increased fourfold to 8. This increase can be attributed to several related factors:

- ⊕ installation of a more user-friendly and powerful messaging system after eight months of system operation
- ⊕ additions of increased functionality in the system as a whole which encouraged employees to use the system more as a tool to do their jobs
- ⊕ expansion of the messaging network due to the addition of new users, inside and outside the company
- ⊕ the learning curve, or period where employees did not simply learn the technical aspects of the system (which are relatively simple) but learned to integrate the system into the execution of their work

There is wide fluctuation in the use of the messaging system both between users and over time. Over the last nine weeks, the mean number of messages per user, per day ranges from less than 3 to over 12 for a given day. It was noted however that there is less individual variation between users during the last period of the study than the early period. As well, wide fluctuations over time for a given user are to be expected as some days users are out of the office, or heavily involved in face-to-face meetings, etc.

Several points regarding command-usage are noteworthy:

- ⊕ The number of commands used per user, per week has fluctuated widely but has shown no marked increase over the period.

- The number of different commands used per user, per week has shown a slow increase as users expand their repertoire of commands.

The main uses of the system appear to be text editing and messaging. During the first 4 months of system operation the average user generated somewhere upwards of 400 pages of textual information. Use of the text editing tool strengthens the hypothesis that managers and professionals will use such a tool, given adequate functionality, rewards and training.

2) Communications

Pretest and posttest information regarding different aspects of the process of communications were collected and analysed. Some noteworthy findings are:

- The percentage of the day spent in communications activities for the Pilot group increased from 53 to 58 percent, while the figure for the Control group decreased from 51 to 45 percent. The reason for the decrease in the Control group is not clear. The increase for the Pilot group is likely due, in part, to a reduction of time spent in administrative, non-communication activities.
- There appears to be a reduction in the use of the telephone and also in one-on-one, face-to-face communications. These findings strengthen the hypothesis that the impact of computer messaging is not simply on replacing paper mail, but rather in substituting for other forms of communication.
- There was an increase in the percentage of an employee's day spent in time which was not an "interruption". This is important data regarding cost-justification of office systems.
- There was an increase in communication among peers. In the pre-study management questionnaire, this was judged to be a desirable potential product of the office system.
- The percentage of attempted communications which failed (e.g., busy phone line) decreased. Reductions in such shadow functions have important and measurable cost-benefit implications.
- Results of subjective evaluations of time use were mixed. There was an increase in the amount of time which was judged to be "useful" but no corresponding increase in the amount of time judged to be "important". Both Pilot and Control groups showed an increase in the amount of time judged to be "time-effective".
- Attitudes towards other employees and communications with other employees improved for the Pilot group only.

3) The Information Resource

It was hypothesized that the disparity between perceived "information needed" and perceived "information received" noted at the time of the pretest, would decrease for the Pilot group. This did not occur. There were a number of improvements between the pretest and posttest in the "information received". However, the perceived "information needed" also increased correspondingly. These findings suggest that as access to information improved, so did expectations regarding what is possible and perceived requirements regarding what is necessary. It is possible that improvements in information were due to a variety of system-related factors including:

- ⊕ access to the information retrieval system
- ⊕ on-line access to the content of previously written reports
- ⊕ access to previous messages
- ⊕ better communications with other employees (e.g., supervisors) who had the desired information

Employees in both groups estimated that they could save substantial time if they had access to a number of specified information topics. It was noted that while projected time-savings are completely subjective and likely over-estimated, two interesting findings stand out:

- ⊕ Both Control and Pilot groups perceive they receive inadequate information and could save about half their time if they had better access to it.
- ⊕ Surprisingly, the Pilot group indicated that they could save more time than the Control group if they had better access to information, even though they were already using an information retrieval system. This likely reflects higher expectations regarding the impact of timely information.

4) Attitudes Towards Electronic Office Systems

It was found that attitudes towards office system technology changed with system-use in several ways:

- ⊕ Attitudes towards the technology and its potential benefits became more positive.
- ⊕ Preference for the use of a keyboard to compose written material increased to the point where users indicated they typed and edited 75 percent of written material they composed, themselves.
- ⊕ There was a positive increase in the opinions of the abilities of fellow workers to use an on-line office system in their day-to-day work.

- ⊕ Attitudes regarding the greatest potential benefits of office systems shifted from an emphasis on clerical, typing and administrative tasks to emphasis on professional time savings and the quality of decisions.

Attitudes towards the specific BNR system were on the whole positive. Users tended to indicate that:

- ⊕ The system is not difficult to use.
- ⊕ They feel comfortable typing at a terminal.
- ⊕ The system is not "down" too much.

Users did, however, express a number of critical attitudes. They tended to agree that:

- ⊕ A better users' manual should be made available.
- ⊕ The error messages of the system are of little value.
- ⊕ The system should be more tolerant of erroneous user input.
- ⊕ Access to a printer is inadequate.
- ⊕ Services degrade during peak workload periods.

Most users felt that the system had improved their ability to do their jobs. Main benefits were seen in the areas of improved communications, better access to information, better preparation of written material, more teamwork, and better collaboration.

Despite positive attitudes, most users felt that the system should be expanded, both in functionality and in the size of the user community. The added functions which received the highest scores were automatic project accounting, computer conferencing and memo and letter templates.

5) *Job Design and Quality of Working Life (QWL)*

It is often assumed that the introduction of computer systems results in a decline in job satisfaction and QWL. Posttest results of the Job Diagnostic Survey indicated the opposite can be true. There appear to be a number of significant improvements for the Pilot group:

- ⊕ the amount of job variety
- ⊕ the amount of significance attributed to the job
- ⊕ a greater sense of accomplishment in the job
- ⊕ a greater desire to be creative
- ⊕ a perceived decline in how simple and repetitive their job is

- ⊕ the amount of personal growth and development
- ⊕ perecived respect and fairness from one's supervisor
- ⊕ the perceived quality of supervision

In general, it appears that the better communications with other employees, less time spent on non-useful activities, easier access to information and so-on had a positive impact on job satisfaction, supervision, employee morale and motivation.

3.1 Conclusion

The study, which has now been extended to other environments, has given some tentative and partial answers to Will Rogers' question: "How can you tell." It has shown that, at least in one implementation, the new office system technology has had a positive impact. Perhaps more important, the researchers have been able to test and refine a methodology for measuring the impact of electronic office systems. As subsequent system implementations are evaluated our ability to collet valid evidence and evidence itself, will grow. If used properly the result should be better-designed office systems -- used, accepted and enjoyed by more office workers.

3.2 Extension of the System

Using the results of the pilot study, it has been possible to make a business case for the extension of the pilot to an Operational System for use by the entire Custom Systems Division. Managers, project leaders and administrative personnel and other employees are being placed on the system, including the President and Vice Presidents who have terminals on their desks.

The Operational system has several additional features. There are user profiles to tailor the user interface to the skill level, requirements and preference of each employee. For example, some users may prefer a menu-based interface. Here menus appear on the screen. When the user wants to do something on the system s/he simply types the number associated with the menu item desired. New users or users who run into trouble are assisted by a Computer Aided Instruction capability.

As well, an Information Management Facility will be implemented in 1981. This provides a set of basic functions for creating database applications. These functions will enable creation and manipulation of databases as well as screen forms display and update. One of the applications to be introduced this year is a project control system based on BNR's Project Management Methodology. This subsystem will assist employees with project financial accounting, tracking, and status reporting.

3.3 User-driven Design

One of the products of the research to date is a methodology for "user-driven design" or customization of electronic office systems to a given office. The method takes information from the user and user environment to

propel each phase of the design process within a specific office. It divides customization of office system into three phases. During the "Prepilot" phase data are collected to identify the most appropriate pilot application, and develop the specification for the system. During the "Pilot" phase information is generated from the system itself and also from the user to help with evaluation and refinement of the system. During the "Operational Office System" phase posttest and system monitoring data are combined with system analysis data to enable the specification and cost- justification of a full Operational System for the organization as a whole.

Using such a method it is possible to more accurately quantify user requirements and to more easily perform cost-benefit analyses for the system selected.

3.4 Ongoing Controlled Field Studies

The methodology developed for and used in this research is now being applied to evaluate the impact of office systems in other environments. With subsequent implementations our ability to collect valid evidence and the evidence itself will grow. This will help to provide the information needed for useful product design, successful marketing and effective implementation of the new technology. The result should be better-designed office systems -- used, accepted and enjoyed by more office workers.

4.0 References

BAIR, James H. "Communication in Office of the Future: Where the Real Payoff May Be." Paper given to the International Computer Communications Conference, Kyoto, Japan, August 1978. p 2-3.

CARLISLE, James. "Evaluating the impact of office automation on top management communication." Proceedings of the N.C.C., 1976.

HILTZ, Roxanne and Murray Turoff. The Network Nation. Addison-Wesley Publishing Co. Inc.,Massachusetts,1978.

TAPSCOTT, Donald. Towards a methodology for office automation communication systems research. Paper presented to the International Workshop on Integrated Office Systems -Organized by the Institut de Reserche d'Informatique et d'Automatique November 6-9, 1979. Versailles France.

TAPSCOTT, Donald, Bharat Shah and David Macfarlane. "Measuring the Impact of Office Information Communication Systems, Vol. 1 - Overview. BNSR Internal Report , 1979.

UHLIG. Ronald P.,David J. Farber and James H. Bair. The Office of the Future. North-Holland Publishing Company. Amsterdam, 1979.

5.0 The Author

Don Tapscott has a Masters degree in Research Methodology. Since 1970 he has conducted research and user requirements studies for a variety of computer systems. He has published extensively on integrated office systems and has spoken across North America and Europe on the topic. Don is currently Project Manager of Office Information Communication Systems (O.I.C.S.) at B-N Software Research, Inc., Toronto.

SECURITY AND MESSAGE SYSTEMS

Chairman - C. Heitmeyer

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
IFIP, 1981

ENCRYPTION IN COMPUTER NETWORKS AND MESSAGE SYSTEMS

Wyn L Price
Division of Numerical Analysis and Computer Science
National Physical Laboratory
Teddington
UK

The protection of data from disclosure or illegal alteration is of prime importance in any communication system, not least in a computer message system where messages await 'collection' in mail boxes. The paper considers the data protection requirements at various levels within the Open Systems Interconnection architecture of ISO and examines also the particular needs of message systems. Both single key and public key cryptosystems are considered, with some reference to the problems of key management within each. The application of cryptographic techniques to message signature is discussed.

INTRODUCTION

With the rapidly increasing reliance placed by commercial and financial systems, eg electronic funds transfer, on data communication networks, it is becoming of very great importance to ensure the privacy and integrity of the transactions which are conveyed. Fortunately the recent and continuing development of cryptographic techniques is of direct application in this context. Both privacy and authentication of messages can be achieved, preventing disclosure and/or falsification of the transported data. It is even possible to devise systems which allow definite attribution of messages to their source, a property which is valuable where contracts may be exchanged via communication networks.

In order that message handling systems should achieve their full potential, it is vitally important that users should have confidence in them. Encryption can play a significant part in establishing the necessary degree of confidence. The same considerations of privacy and integrity apply in this context as in the wider field of communication networking in general. If users are to exchange encrypted messages, then means must be found of establishing common encryption algorithms and compatible keys. Digital signatures may be employed to provide acceptable proof of message origin.

This paper examines ways in which cryptographic measures can be applied in the communications context in general; we then consider the particular needs of computer message systems.

To some extent we shall relate our discussion to the International Standards Organisation (ISO) open systems interconnection (OSI) model (ISO (1979)), which is intended to provide an architectural framework into which work on network communication protocol standards can be fitted. Seven architectural levels are specified; these are named and numbered as shown in figure 1. The OSI structure is intended more as a guide than as an authoritative taxonomy of functions. In some contexts it is possible that one or more levels may be omitted from a particular implementation. Note that figure 1 also indicates at each level the extent to which encryption may be applied.

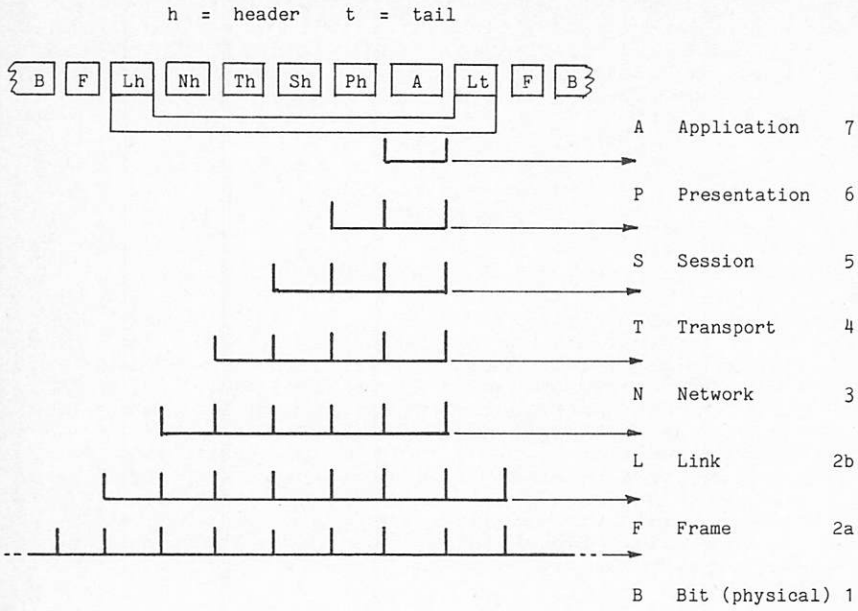


Figure 1 Architectural levels of the OSI model, showing the extent to which encryption may be applied at each inter-level interface.

In order to achieve protection of data carried on inter-node links or held in temporary storage within network nodes, cryptographic devices can be installed within the subnetwork (the nodes and the links joining them) itself. In terms of network architecture this covers the three lower levels of the OSI model. The upper three or four levels of this model include services which are oriented more to the user than to the communication network. Encryption may also have an important part to play at these higher levels.

Much work has already been done on defining standard protocols whereby users may communicate accurately and efficiently via data communication networks. Levels 1 to 3 (physical, link and network) of the OSI model are already well defined. Level 4 (transport) protocols are in course of preparation. Much remains to be done on levels 5 to 7 (session, presentation and application). At least two of the draft definitions of level 4 include references to the provision of encryption as an optional enhancement to a transport service, but as yet this enhancement remains to be defined. The OSI document itself makes a brief mention of encryption at level 6, presentation level. [Note that the distribution of functions between levels 3 and 4 is undergoing changes at ISO at the time of writing. The present paper assumes the status quo before these changes, but the conclusions drawn are unlikely to be seriously affected.]

In the field of proprietary networks, IBM have included well-developed cryptographic facilities in their SNA architecture (Ehrsam et al (1978)); these are provided at session level. They allow, inter alia, for a sophisticated system of cryptographic key management. We shall see that key management is an important subject requiring careful attention in a wider context. The IBM scheme is well suited for SNA, though it may not necessarily provide a pattern on which general secure network design principles can be founded.

Clearly, if users are to communicate one with another using cryptographic techniques, then they must use compatible cryptographic systems. A cryptographic system includes the definition of the encryption algorithm(s) and the rules (protocol) by which it is to be used. A multiplicity of different encryption algorithms is unlikely to be tolerable. A prime candidate for general adoption as an encryption algorithm is the US Federal Data Encryption Standard or DES (National Bureau of Standards (1977)), though the rules for its use are still in course of development. This algorithm is already being considered by ANSI (X3T1 and X3S3.8) and by BSI (OIS/21) as the possible basis of national standards; international consideration has commenced in Working Group 1 of ISO Technical Committee 97. However, other encryption algorithms have properties which may make them particularly appropriate for use in certain communication contexts. In particular we have in mind the general class of public key cryptosystems. The ability of some of these systems to provide convenient digital signatures on transmitted documents has important potential in the conduct of electronic commerce. Furthermore, the key distribution problem takes on an entirely new guise in this context. If these alternative algorithms are to receive wide adoption, then standardisation activities must soon commence in this field also.

We shall see that there are a large number of places in which encryption facilities may be sited within a complete communication system. Clearly it is neither necessary nor economic to introduce encryption at all these potential locations. Encryption almost always introduces overheads, in terms of additional procedures, not to mention the additional cost of the encryption system itself. Therefore the points at which encryption is introduced must be limited and well chosen.

In what follows we shall first be concerned with "single-key" encryption, where the encryption algorithm may be placed in encrypt or decrypt mode, but needs the same key in each mode for correct operation; the DES is an example of such an algorithm. We shall later look at public key cryptosystems, which are not of this type.

ENCRYPTION IN THE COMMUNICATION SUB-NETWORK

The simplest point at which to introduce encryption devices within a packet-switched subnetwork is between node and line modem; in OSI terms this could be between levels 1 and 2. In this configuration the whole packet, header and data, may be encrypted, transmitted in encrypted form and then decrypted on receipt before processing it at the next node; this mode protects data on the line. An intruder using a line-tap can neither interpret the data nor deduce its source or destination. If a pattern of dummy traffic is kept up when the line would otherwise be idle, then even the most primitive form of traffic analysis is frustrated. However, such an installation gives no protection to packets held in temporary storage within a node. Nodes notoriously go wrong on occasion and may misdirect packets. A bug introduced into node software may cause a deliberate breach of security such as the copying of packets to illicit destinations. Encryption installed at this point affords no protection against disclosure of a packet within a node nor against accidental or deliberate misdirection. However, if the communication links are of a particularly vulnerable type (eg microwave links), then this kind of encryption may be worthwhile. We shall also see that it may be used profitably in conjunction

with other encryption modes.

Enhanced protection may be achieved by transferring the encryption devices from the line to a point within the node itself. Part of the node may be set aside as a security module and the packet data allowed to appear in plaintext only within this module. Security then depends vitally on the strength of the module. Decryption and encryption of the data take place within the module because a different key is used for each new link traversed; the module holds a list of current keys for connected links. The network level packet header is left unencrypted in order that the general node functions, such as packet routing, may be carried out. This mode of operation has been called node-by-node encryption and would be located either between OSI levels 2 and 3 or within level 3.

Undoubtedly the greatest subnetwork security is achieved if the encryption function is arranged on an end-to-end basis. In this mode the encryption devices are located, for example, near the user's terminal and near the host computer. Only the data is encrypted and the header is sent in clear form. Thus an intruder, tapping a line, can read the source and destination of packets, but cannot read the data. A combination of end-to-end encryption and encryption at the physical or line level would frustrate the reading of addresses, whilst maintaining data security within the network nodes. A node bug would yield, at most, only the traffic addresses. Key management in end-to-end encryption presents an additional task for the system; the communicating parties must possess the same key and this may not be easy to arrange in a large network or across several inter-connected networks.

It is a moot point in regard to end-to-end encryption whether this should be provided by the network users or by the network authority. Since the X.25 interface is simply a means of connecting DTEs to DCEs, with incomplete end-to-end significance, it is not possible just to add encryption to a DTE with an X.25 connection. On the other hand, cleartext data may be accepted across an X.25 interface by the DCE from the DTE and encrypted on an end-to-end basis by an encryption function contained within the DCE, being decrypted by the destination DCE and delivered in cleartext form to the destination DTE. This configuration has been put forward as a possibility by Nelson (1979). It is not clear whether users will be content to entrust their confidential data to an encryption service whose integrity depends on the correct behaviour of the network authority over which the user has no control. Used by itself the method also has the weakness that the transfer of data between DTE and DCE and vice versa is not protected by encryption.

ENCRYPTION AT THE TRANSPORT SERVICE LEVEL

The highest level of the OSI network architecture specifically devoted to the communication service (and therefore the highest level likely to be provided in normal circumstances by the network authority) is the transport service (level 4). Above this the architectural levels are more particularly angled to the needs of the user. The aim of the transport service is to allow transparent transfer of data in a reliable and cost-effective manner between the session entities located in the layer above, relieving these entities of any concern for the detailed way in which this is carried out. It is at the transport service level that end-to-end call control is available on a fully developed basis. Therefore this level is a natural candidate for the location of an encryption facility, even though the service may possibly be provided by the network authority. The encryption function may be provided within level 4 at one of a number of alternative locations - at the interface with the session level (5), within the body of level 4 or at the interface with the network level (3).

Encryption provided by the transport service will be applied in response to a request from the session level for an encrypted call. Encryption may also be

imposed because the distant session entity will only accept a call if this is encrypted.

The transport service peer protocol provides for a negotiation phase in which desired call parameters are offered to a distant location; the latter may either indicate acceptance of the call as offered, it may reject the call or it may offer to accept the call provided alternative call parameters are substituted. A negotiation phase is a very important part of the protocol of any transport service encryption enhancement. The potential communicating parties must achieve agreement on:

a) the encryption algorithm to be used
b) the mode in which the algorithm is to be used
c) the encryption keys
d) the initialisation parameters.

Exchange of encryption parameters may begin during the call negotiation phase, but it is unlikely that agreement on the encryption parameters can be completed during normal call negotiation, because encryption is an enhancement at level 4; the encryption negotiation protocol does not form part of the connection protocol. In other words encryption negotiation can only be completed after a call has been set up. At this stage the call is unencrypted. If the encryption negotiations are successfully concluded, then the call will be switched into an encrypted mode. If encryption negotiation fails, then the call will be closed down without delay.

It is feasible, without loss of security, to negotiate the particular algorithm and the mode of its use in a fashion which can be overheard by opponents. However, encryption keys and initialisation parameters must not be exposed in this way. If single key cryptography (eg the DES) is to be used, then this requires that some prior agreement, not conducted via the network, must exist between the potential communicating parties. Each must possess a common encryption key (eg a master key), under whose protection session keys and initialisation parameters can be exchanged. An alternative is to prepare a list of session parameters for later use, this list to be held at each location. An index reference to the list, exchanged during the encryption negotiation, indicates the parameters to be used in the current call. Possession of secret information in common implies some secure form of data transport, possibly requiring a personal visit of a security officer to each site. In an open networking context where a user may wish to communicate securely on occasions with any of a large number of other users, this is not a comfortable situation. We shall return to this point when we consider key management systems.

Once a secure call has been set up, the integrity of the transferred data must be assured. The transport service encryption enhancement must check that the received encrypted data stream is the same as that transmitted from the source. In order to ensure that no data has been altered, deleted or inserted, some form of cryptographic data chaining must be employed. In the DES context this means that ciphertext feedback or cipher block chaining modes of use must be employed. Furthermore, successive chains must be linked together by a system of chain identification numbering which is contained within the encrypted data stream. The transport service protocol does not guarantee delivery of data, but it does guarantee that data delivered will be in the correct sequence as it is offered to the session entity. If a malfunction of the call happens, for example if a Reset occurs, then some data may be lost. Loss of data should be reported to the session entity on whose behalf the transport connection is operating; this entity should decide on the point from which retransmission should commence. Entire retransmission of the data of an encrypted call should be avoided on efficiency grounds; retransmission of the same data with different encryption parameters must be avoided on the grounds of cryptographic security.

When the stream of encrypted data is ended, the call must be closed down in an

organised fashion. It is important that the Disconnect message is sent only after the final delivery of data to the destination (including the complete cryptographic integrity check, in case some retransmission is needed). Otherwise data may be overtaken by the Disconnect message and lost, making it impossible to complete the check correctly. Once it is clear that the data transfer is satisfactorily completed, the call should be closed down in the normal way, but the encryption enhancements at both source and destination should overwrite any current working keys and initialisation parameters with random data. Nothing should be left that would enable an opponent at some later time to make sense of encrypted data recorded in transmission somewhere within the network.

ENCRYPTION AT HIGHER NETWORK LAYERS

We now consider the provision of encryption at the session, presentation and application layers of the network architectural model. These layers are in the user domain and it may be considered that encryption provided here will lead to a reduction in the risks of exposure of plaintext to disclosure or alteration. The higher the level, the less the degree to which facilities are shared. However, it is only the data which may be encrypted at the higher levels, not the "envelope" in which the data travels, see figure 1. The higher the level at which encryption is installed, the greater the exposure of the envelope information. An opponent can expect to be able to read control data such as source and destination process identifiers, size of data, etc, etc, from which useful insights into user traffic may sometimes be gleaned.

It can be argued that provision of encryption above the session layer is pointless. The session layer is under user control and therefore the user has considerable assurance of system integrity if encryption is installed here. However, a session layer may work on behalf of different users; separation within the layer must be ensured. Since the presentation layer is concerned with such matters as data formatting, encryption of data at the application layer would seem to conflict with the presentation layer function. The task of the presentation layer itself is sufficiently complicated without burdening it with encryption responsibilities. This leaves us with the session layer, whose defined task is not onerous; it seems sensible to install an encryption function here. Only for data of a very high security rating should it be necessary to include cryptographic protection at a higher architectural level. Indeed many applications processes handling data of this type already include their own data protective measures; further protection from network facilities may not be necessary.

Concerning the design details of encryption at session level, these could be very similar to those we have outlined for the transport level.

KEY MANAGEMENT

We have now discussed the installation of encryption devices and functions at various network levels. The devices we have been considering have been of the single key type, where communicating devices have to be loaded with identical keys before meaningful data exchange is possible. Ensuring that the right keys are available in the right places at the right times and that they are not exposed to disclosure is no mean task.

If line bracketting encryption devices are installed on lines between network nodes, then local operators could be made responsible for installing communication keys. Secure down-line loading of keys from one device to the next is possible if a system of master and secondary keys is operated.

Master keys may be installed at transport service or session layer encryption

facilities by local operators. Session keys with life limited to call duration can be generated by local software. However, this kind of approach to key management leads to a very fragmented solution, placing responsibility for system security on too many individuals and system components; it goes no way towards solving the problem of enabling secure communication between any pair of a large number of users connected to a data network or system of connected networks.

If this is possible (eg when a network is used only by a mutually trusting group of users), it is far better to vest the control of key management in a single network location (with backup to allow for possible failure). This can be provided by a special purpose host computer attached to the network; let us call this host a network security centre (NSC). Even when the network users do not all belong to a mutually trusting group, there is much to be said for establishing a network security centre.

If link encryption is installed, then a "ghost" host (a key controller) at each network node can be given responsibility for controlling the keys of local network lines (in collaboration with the key controller in neighbouring nodes). When a new key is required for a link, this is requested by the local key controller from the NSC, using a communication key held in common by the key controller and NSC to protect the transfer of the new link key. The unique key controller/NSC communication key must be loaded at each node by a travelling security officer. Since this key will not be used very frequently, and then only for exchanging keys, it will not need changing very frequently.

Similarly, for example, a transport service encryption facility may communicate with the NSC using a communication key unique to that facility, requesting a session key for communication with a remote destination. This new key will be returned to the source transport service under encryption and sent at the same time to the destination transport service under encryption of the communication key unique to that transport service. By this means a session key may be set up at the two ends of a call. The call encryption negotiation can be designed to check that the same key exists at both transport services. Variations on this scheme have been suggested by Heinrich (1978).

When the communicating users are connected to two different, but interconnected, networks, two network security centres may be involved. The allocation of keys will then be based on control shared between the centres; an inter-NSC protocol must be provided. Encryption should still be on an end-to-end basis; separate encryption within each network accompanied by plaintext handover at an inter-network gateway must not be entertained.

In the IBM key management scheme the control on keys is vested in each host computer; there is no separate network security centre. Master keys exist at the host in order to protect terminal keys, file keys and inter-host keys for file transfer. An extensive range of encryption functions allows only those key translations that are legitimate in each context.

Our discussion of key management has assumed so far that the encryption function is carried out using single key cryptosystems. The situation may change totally if we introduce the possibility of using public key cryptosystems alongside the single key systems.

PUBLIC KEY CRYPTOSYSTEMS IN NETWORKING

The concept of public key cryptosystems (PKCs), introduced by Diffie and Hellman (1976), is now widely known. Several practical schemes (Rivest et al (1978), Merkle and Hellman (1978)) have been proposed. It is likely that at least one of these systems may be expressed in special purpose LSI hardware, thereby overcoming some of the disadvantage that most PKCs suffer from, that of imposing

a heavy computational load.

The fundamental principle of PKCs is that data is encrypted under a public key peculiar to the intended recipient; only the latter, in possession of the corresponding secret key, can decrypt the received message. Thus, in order to communicate with a particular destination, a source need only know the public key associated with that destination. No longer is there any need to set up a call with the same key available at both ends. The cryptographic negotiation phase of a transport service encryption enhancement need only check with the destination that a particular PKC algorithm is acceptable and obtain the required public key. Evidently it is vital that the source from which that key is obtained is completely reliable. The function of holding public keys and making them available on request can be allocated to the NSC. Secure means, almost certainly involving direct human action, must be provided for notifying public keys to the NSC. Any user whose key is so held would be well advised to check from time to time that the NSC still holds his correct public key. Otherwise an opponent could substitute a false public key to which he holds the corresponding secret key, enabling him to decrypt any data sent so encrypted. Naturally the rightful recipient would be unable to decrypt the data and so might become quickly aware of the substitution.

If it should be considered that, despite advances in hardware design, PKCs are still computationally too inefficient for data encryption, then it is quite possible for the encryption enhancement to employ a PKC to transport securely a session key for use in an efficient single key encryption system such as the DES.

AUTHENTICATION, IDENTIFICATION AND SIGNATURES

In order to give added assurance of communication integrity, it may be desired to add an authentication code to a message, whether it be encrypted or plain-text. What is needful is the result of some one-way function of the message, usually much shorter than the message itself; an encryption function can conveniently be used to generate such an authentication code (Meyer and Matyas (1981)). Any change to the message affects the authentication code in an unpredictable fashion, so that it is infeasible to make compensating changes to message text and authentication code. The authentication code is generated at source, sent with the message, regenerated by the recipient and compared.

Positive identification of a distant entity with which communication is desired is not easy to achieve. Prior agreement with a distant user (not via the network) on the exact keys to be used in an encrypted call enables a check to made on the presence of the agreed keys, testing that the distant user is the person with whom the agreement was made. Identification then hinges around the security with which these keys are protected both in use and in storage. Keys used for this purpose are outside the scope of the kind of key management system discussed earlier. Positive identification of a distant user may be made more certain by checking on personal characteristics, eg signatures, fingerprints, voiceprints, but the message exchange that establishes this identity must itself be safeguarded by encryption and authentication. A secure central register of personal characteristics would be helpful in designing such a system, otherwise each user must maintain a file of characteristics of desired correspondents.

The concept of a digital signature attached to network messages is very important. The secure conduct of business via communication networks demands that proof of origin of messages shall be available. Once again, certain kinds of encryption can be deployed for this purpose. Single key cryptosystems are not ideal, since the receiver, being in possession of the key, could easily generate a ciphertext to correspond to a forged plaintext message. A sender, wishing to repudiate a message that was actually sent, need only claim that the receiver forged it. Various rather complex and inconvenient schemes (eg Lipton

and Matyas (1978)) have been devised in order to apply single key cryptosystems to this need. Those public key cryptosystems whose crypting/decrypting transformations can be applied to text in either order lend themselves conveniently to signature generation (Davies and Price (1980)). A plaintext message transformed by the sender using his secret key information can be restored to plaintext by the recipient using the public key corresponding to the putative sender, thus identifying the source unequivocally. Clearly the latter must be identified in a plaintext message header in order to select the relevant public key. Such a message is, of course, intelligible to all parties connected to the network, but secrecy can be restored by also encrypting under the public key of the intended recipient. In order to distinguish messages from others sent previously between the same pair of users, some information such as date and time of origin must be included in each message; this is in no wise different from the way in which commercial orders are dated and numbered when sent by conventional means. In cases where a dispute arises as to the true origin of a message, then some kind of arbitration is necessary. This arbitration must have recourse to the key register at the NSC, but the latter should not be expected to provide judgement. This must be reserved, possibly to some quasi-judicial authority, independent of the network authority.

ENCRYPTION IN COMPUTER MESSAGE SYSTEMS

Broadly speaking, encryption of data in a message system can be organised in either of two ways. In the first, messages are encrypted under the control of a sender process (located, say, at level 5, 6 or 7) to be sent via the message system and placed in the receiver's mailbox, there to be retrieved and decrypted as required, using an algorithm residing in a receiver peer process; the mailbox is under the direct control of the receiver. In the second, three successive encryption operations are involved; a local host handles the mailbox in each case, communication with the local host being encrypted under some encryption algorithm and key; re-encryption with a new key (and possibly a different algorithm) takes place in the sender host before the message is sent to the receiver host; finally the message is again re-encrypted at the receiver host into a form which can be decrypted by the receiver. It is assumed that each host provides a number of users with mailbox services. The IBM encryption system under SNA could be used in this mode for encryption in a message system. Stringent precautions (such as are available in the IBM scheme) are necessary to prevent disclosure whilst the message resides in the host. In the first method, described above, the message is not subject to re-encryption during the transit from sender to receiver, which may be considered to be a more secure mode of message handling; on the other hand a greater responsibility and work load is placed on the user.

Whichever method is used, of its very nature a computer message system does not complete the entire message transport during one session or call; there is therefore no call parameter negotiation phase between sender and recipient, and, in particular, no cryptographic parameter negotiation. Nevertheless, for successful encryption, there must be consistency between sender and receiver on the algorithm and parameters to be employed; this consistency must exist in the first method, above, directly between sender and receiver; in the second method, the consistency must be between the respective host computers. Either a prior (and non-ephemeral) agreement must exist between the parties, or an encryption method must be used which does not depend so heavily on prior agreement. From our earlier discussion of single key and public key cryptosystems it will be apparent that single key systems, used by themselves for message exchange, would need prior agreement. This is undesirable, as it effectively gives longer currency to what is to all intents and purposes a session key, possibly making the task of a cryptanalyst somewhat easier. On the other hand the public key cryptosystem requires only a reference to a central register of public keys; this should confirm the cryptographic algorithm in use by the intended message recipient and will yield the latter's public key.

Secure messages can now be sent without either prior agreement between correspondents or a cryptographic negotiation phase. The recipient can decrypt at his leisure messages awaiting his attention.

If it is desired to employ the more efficient DES encryption algorithm, then a DES encrypted message may be accompanied by the appropriate DES key (generated randomly for this message) which is itself protected by a public key encryption. The recipient first obtains the DES key by using his secret (PKC) key and then decrypts the message using the DES algorithm in the normal way.

If the correspondents are respectively attached to different computer networks, then some communication between network security centres may be required in order to establish the parameters of the intended recipient. The same strictures apply regarding the avoidance of any decryption and re-encryption at network gateways.

Signatures can be obtained in a computer message system context in exactly the same way as in the more general sense already discussed.

CONCLUSIONS

Encryption devices may be installed within communication systems at many possible places. Over-provision of encryption is expensive and should be avoided. Adequate security may be achieved by providing encryption at levels 4 or 5 of the OSI architecture, with the addition of encryption between levels 1 and 2 if traffic security is required.

Key management may require establishment of a network security centre, which also has an important role as a key register if public key cryptosystems are operated. The latter may afford the possibility of digital signatures, which could be very important in commercial applications.

Where network interconnection is involved, each network may have its separate NSC; a secure inter-NSC communication protocol will then be necessary. Messages sent between networks should not be decrypted and re-encrypted at network gateways.

In the particular context of computer message systems, public key encryption is particularly convenient, since it avoids any need for prior agreement between correspondents. It is only necessary that all potential correspondents shall register their chosen algorithm and encryption parameters with a network security centre. The ability to provide computer messages with digital signatures is potentially very important.

REFERENCES

[1] International Organisation for Standardisation. Open Systems Interconnection. Document ISO/TC97/SC16 N227 (October 1979).

[2] Ehrsam, S.M., Matyas, S.M., Meyer, C.H. & Tuchman, W.L., A cryptographic key management scheme for implementing the Data Encryption Standard, IBM Systems Journal 17 (1978) 106-125.

[3] National Bureau of Standards. Data Encryption Standard. NBS FIPS Publication 46 (January 15, 1977).

[4] Nelson, J., Implementations of encryption in an "open system" architecture, Computer Networking Symposium, National Bureau of Standards (December 1979).

[5] Heinrich, F., The network security center: a system level approach to computer network security, NBS Special Publication 500-21 Vol. 2 (January 1978).

[6] Diffie, W. & Hellman, M.E., New directions in cryptography, IEEE Trans. on Information Theory IT-22 (November 1976) 644-654.

[7] Rivest, R.L., Shamir, A. & Adleman, L., A method of obtaining digital signatures and public-key cryptosystems, Comm. ACM 21 (February 1978) 120-126.

[8] Merkle, R.C. & Hellman, M.E., Hiding information and signatures in trap-door knapsacks, IEEE Trans. on Information Theory IT-24 (September 1978) 525-530.

[9] Meyer, C.H. & Matyas, S.M., Cryptography - a new dimension in computer data security. (To be published by John Wiley & Sons, New York.)

[10] Lipton, S.M. & Matyas, S.M., Making the digital signature legal - and safeguarded, Data Communications (February 1978) 41-52.

[11] Davies, D.W. & Price, W.L., The application of digital signatures based on public key cryptosystems, Proc. 4th ICCC, Atlanta, Ga., (October 1980) 525-530.

HUMAN FACTORS IN COMPUTER MESSAGE SYSTEMS

Chairman - G. Attardi

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

THE USER INTERFACE DESIGN PROCESS

Hilary Williamson
Sabine Rohlfs

Messaging Systems Department
Bell-Northern Research Ltd.
Ottawa, Ontario
Canada

This paper describes a pragmatic approach to user-oriented systems design. General phases in the design of a user interface are presented. Design phases and design justification are illustrated by examples from an organizational messaging system.

1. INTRODUCTION

The success of new office products and value-added network sevices will depend not only on their functionality, but also on their ease of use and user acceptance of them. To a great extent, these determine the efficiency and effectiveness gains - and therefore the success - of a system. This implies serious attention to the user interface.

User interface design traditionally refers to that subset of the system design process which deals with the layer of functionality driving the user - system interaction. Historically this was done by computer scientists as one component of an overall system design. This led to a computer-oriented design. More recently, there has been a growing interest from ergonomics and psychology in making the design more user-oriented.

When the user interface designer delves into the existing literature he finds: description of hardware such as lightpen or softkey and how they can be used to facilitate user - system dialogue [1]; general guidelines for system behaviour and the role of user and system in a dialogue [2], [3]; do's and don'ts for special purpose systems such as data entry systems [4], and classification schemes by task, type of problem solving activity, type of terminal, and so on [5], [6], [7].

What he does not find is a methodology that combines traditional systems analysis with the more recent guidelines from other disciplines. This makes it particularly difficult to design a system such as an office system, where there is a divergent user population with different user and organizational requirements, and where neither the technology nor its impacts are well understood.

This paper suggests a pragmatic aproach, based on a methodology for user-oriented design. This approach is based on the authors' experience in various projects, and some evidence from the literature. The paper does not cover all aspects of system design, nor does it consider the implementation phases.

Section 2 of the paper examines the user interface designer's position within the organization responsible for the product. An electronic messaging system is introduced in section 3. Phases in the design process are presented and illustrated by examples from the messaging system in section 4. The paper concludes with some general recommendations.

2. THE DESIGNER'S DILEMMA

During the design process the designer often finds himself in the position of having to find a compromise between the contradictory requirements of different users - one user's "must have" is another one's "not needed". Traditionally, designer and user have been described as a sort of Odd Couple (see for example [8]).

Taking a broader look, there are a variety of groups with influence on system design: for example, an organization might involve its marketing group, engineering and product control in addition to the designer's own management, and a steering committee. With this perspective, the Odd Couple expands into an Odd Quintuple:

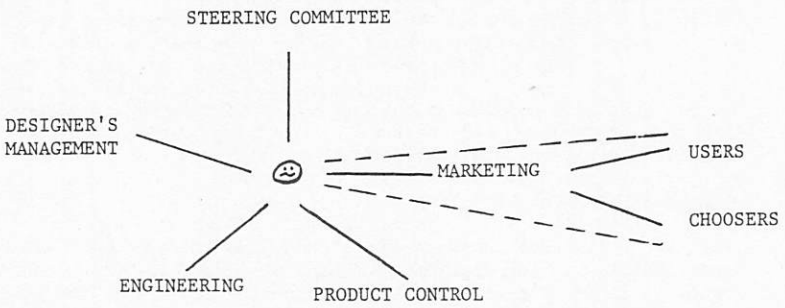


Figure I: The Odd Quintuple

The groups involved will vary from one organization to another, depending on organization structure, and the type of project and product management involved [9]. The picture becomes more complex when the design is for a product rather than an in-house system. In the case of a product it is sometimes difficult to identify future users, and hence participative design techniques (as described in [10]) are not directly applicable. The figure above illustrates both cases. In the in-house situation the designer has direct contact with users. In the case of a product, marketing often represents the end user, and are the designer's customer.

Each group in the example has a different perspective on the system, and different measurements of reward in their own positions. As a result, they often pull the design in different directions.

Marketing's view of the system is oriented towards the needs of their current and anticipated customer base, potential revenue and competitive position. Their view is in turn influenced to a large extent by the organizational requirements of the potential buyers, and less by the needs of the individual user. The choosers make

the decision to purchase, but are rarely users themselves [11].

The product control group is interested in overall product integrity and in administration. Their bias is towards ease of support and maintenance, extensibility and release control. The emphasis is on administrative tools and interfaces, which become their measure of success.

Engineering are interested in a design that facilitates development, and helps them to control cost and schedule. Their aim is often ease of implementation, which does not necessarily imply ease of use. They are prone to resist changes in the design, particularly ongoing ones.

The designer's management is often interested in proving that they are on top of state-of-the-art developments in the field. This is their measure of success.

The mandate of the steering committee is to ensure design compatibility with company policy, judge the quality of the work, and arbitrate conflicts [12]. But it is hard to judge design quality [13], and experience shows that committee members are often influenced by their own background and positions.

Thus, the designer is caught between conflicting goals. This results in a bargaining situation, with the designer striving to hold together a consistent design, and to keep everyone happy. This often results in a design that satisfies no one.

The designer must understand these conflicting demands and pressures, and try to generate cooperation and consensus. The steering committee takes on this role for high-level decisions, but the designer himself must negotiate towards consensus on the myriad of detail decisions that determine a user-oriented design. He has to work as an educator, facilitator, consultant, and advisor, without taking the role of an arbitrator [14]. The designer must face the fact that design is as much a political as a conceptual process. "Unfortunately, 'politics' have been equated with evil, ... but politics are the process of getting commitment, or building support, or creating momentum for change." [12].

The intent of the remainder of this paper is to provide some guidelines for the conceptual side of the design process, and some techniques for justification of design that can aid the designer in consensus generation.

3. OVERVIEW OF A MESSAGING DESIGN

The following discusses design considerations for an organizational messaging system, and introduces an organizational messaging model. This is then used as a basis for presentation of phases in the user interface design process.

3.1 Organizational Messaging and Its Implications for Design

Today's perspective on messaging has emerged out of two very different backgrounds. The earlier public message switching services cater to the point-to-point transmission of textual information, and are used for the exchange of such information as stock lists or pricing updates. They are mainly used by the business community for "administrative" messaging purposes.

The later electronic message and conferencing systems, such as MSG [15], HERMES

[16], FORUM [17], EIES [18], were initially developed and used by a select scientific community as a replacement for certain types of phone calls, letters and face-to-face meetings. These systems are largely used for communication of "personal" messages between individuals.

Not only do these existing message systems come from historically different backgrounds, but they also serve very different user needs.

A system for general use within an organization should support both personal and administrative messaging features. In some respects, administrative and personal messaging represent extremes of usage:

> Administrative messaging is more formal. Communication paths tend to be fixed and related to the structure of an organization. Communication events are driven by organizational procedures. An example would be daily price updates from a head office to all branch offices, prepared by managers and entered by teletype operators.
>
> Personal messaging tends to be informal. Communication paths are more dynamic, from individual to individual. Communication events are based on need, which varies from day to day. As an example, John Smith asks Ann Dupont about her personal opinion of the outcome of a meeting.

Organizational messaging incorporates both administrative and personal usage. As an example, a manager sends monthly status reports, and also communicates with various people on a day-to-day problem solving basis. These two extremes in the spectrum of usage lead to design emphasis on flexibility in the basic functions.

3.2 Overview of an Organizational Messaging User Interface

A messaging user is an individual (or a group with the same job function), with a single user identity. A messaging user accesses the system interactively to read and send messages. This is the most common form of access for personal messaging use.

Another form of access, that originated in administrative systems, is provided by the auto-delivery station. This is a terminal that can accept delivery of messages in unattended mode, on behalf of one or more messaging users. For administrative messaging, this provides an alternative to signing on to read mail. In personal messaging, it might be used simply to provide a printed record of mail, or a general delivery facility.

An organization is a named set of messaging users and auto-delivery stations. An organization has an organization coordinator. This is a messaging user who has additional capabilities to administer the users of an organization.

Every messaging user has a basic environment that resembles his desk-top:

> an in-basket for his incoming messages with a tabular overview in an in-list,
>
> an out-basket containing certain outgoing messages with an overview in an out-list,
>
> a message sheet for message composition,

a note-file as a private filing area, with an overview in the note-list,

a personal profile.

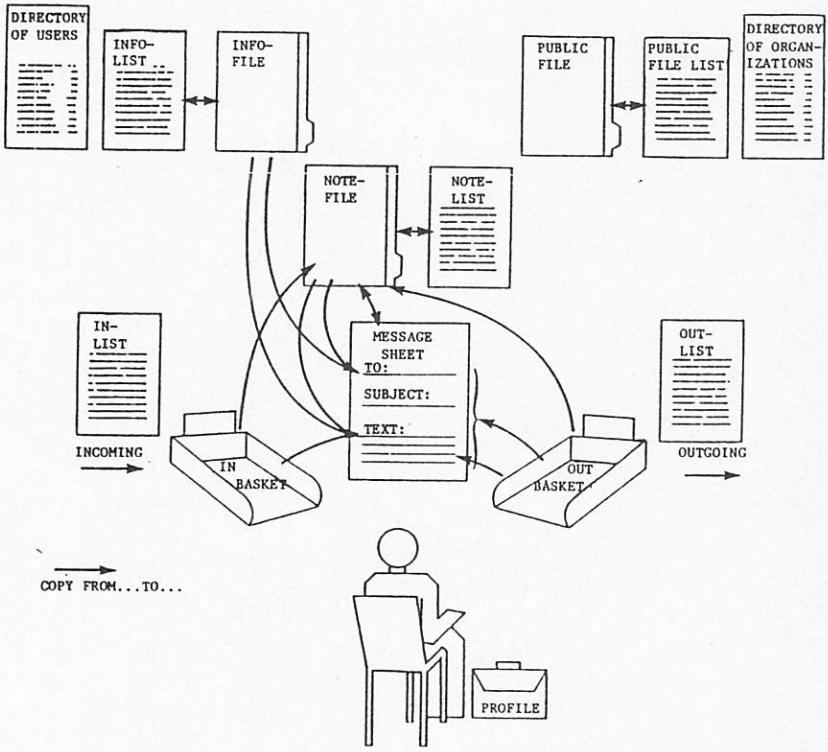


Figure II: User Environment

The elements of a user's messaging environment and their inter-relationships are described in the following.

The in-basket is a repository for all user's incoming messages and serves the same purpose as the in-basket on the desk-top. The in-list contains an overview with one entry per message, in order of arrival. Each entry is referenced by its temporary sequence number, a tag provided for ease of issuing commands. In addition, the system assigns a unique, system-wide message identification to every message when it is sent. This global identification can be used in dialogue between users.

Each in-list entry contains two flags. The delivery flag highlights special cases in message delivery, such as that the message is urgent, or has been automatically

forwarded to this user.

The status flag tells the recipient how much he already knows about the message, in order to facilitate use of the in-list as a "to-do" list. For example, the flag can be:

-"new" for a newly arrived message,
-"listed" if the user has seen its in-list entry,
-"read" if the user has read it,
-"auto-delivered".

The user also has a wastebasket for messages that are no longer needed. In the office environment, papers can be taken out of a wastebasket, uncrumpled and dealt with again, if they were thrown away by mistake. Similarly the user can crumple a message into an electronic wastebasket and uncrumple it before the wastebasket is emptied.

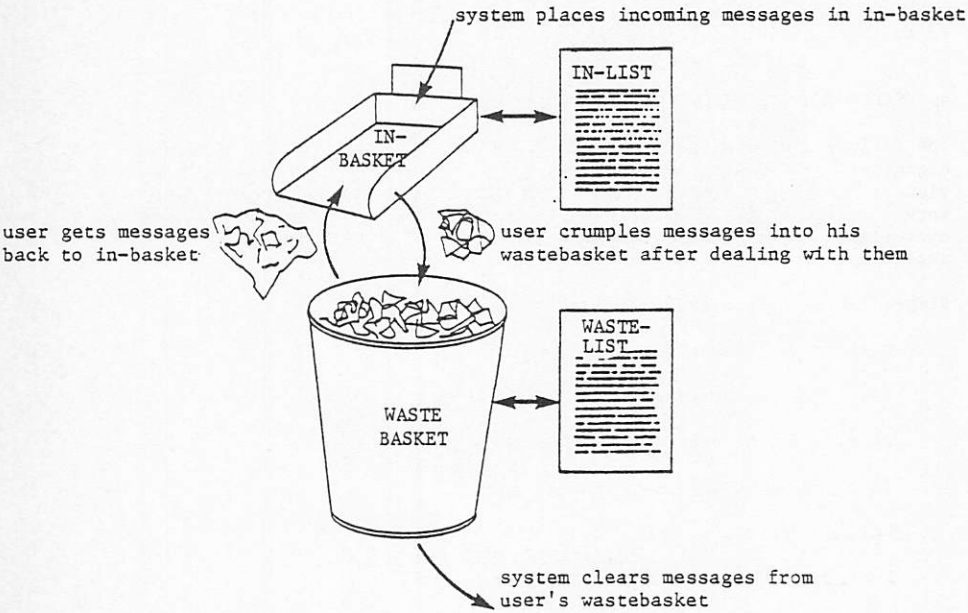

system places incoming messages in in-basket
IN-LIST
IN-BASKET
user gets messages back to in-basket
user crumples messages into his wastebasket after dealing with them
WASTE-LIST
WASTE BASKET
system clears messages from user's wastebasket

Figure III: Wastebasket

In the standard office environment, the out-basket is a holding area for outgoing mail. This does not apply to the electronic out-basket since the system dispatches messages immediately. Instead, this out-basket can be used as a holding area for messages being "tracked", and the out-list becomes a "to-follow-up" list.

Tracking of delivery is critical for administrative messaging use when there is a time limit on important information, as for sensitive pricing data. It is also useful in personal messaging, for example in setting up an urgent meeting. This is discussed in more detail in section 4.3.1.

Message composition is based on the message sheet, which is like a sheet of formatted stationary always available on the desk-top. Its fields (including the recipient list) can be entered in any order, and the message sheet edited as a whole, just as for a conventional letter.

A user can compose a message by issuing commands to fill the specific fields of the message sheet, or he can be prompted to enter them by the system. He can copy a distribution list to the To: field and then insert more addresses. He can copy items from a private file folder into the contents field. He can copy entire messages from the in-basket or out-basket and insert comments, in order to annotate and then forward the information.

A user's personal profile provides a means of tailoring usage to his own needs. The organization coordinator sets initial profile values when he adds a user to the system, and only he can change certain profile permissions that are under organizational control. The user himself can change any other profile values at any time. A user need not bother with his profile but, with experience, he may wish to adjust his mode of interaction.

4. USER INTERFACE DESIGN PHASES

The following discusses phases in a user-oriented design, in the context of the organizational messaging system described above. This presents the authors' own view of the design process, based on the messaging design above and on previous work ([19], [20], [21]). It does not attempt to cover all phases in the product cycle from market research to implementation, but only the user-oriented design aspects.

Stages in the design are:

Goal setting based on requirements.

Design of a functional model.

Development of scenarios of use.

Identification of dimensions of use.

Design of feature sets.

Personalization of features.

User documentation.

Finalization of the interaction language.

As in any other design effort, these stages are iterative - the diagram below only reflects major iterations. The designer must communicate his design, and the focus for that communication changes throughout the design process towards different groups in the organization.

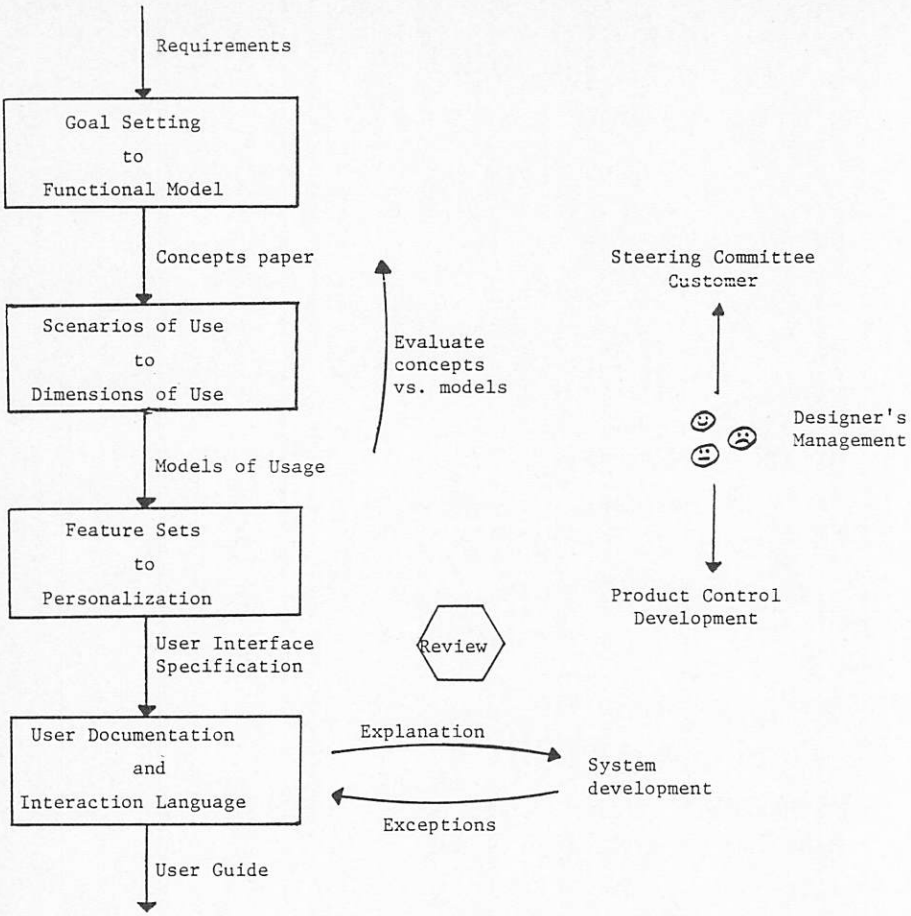


Figure IV: Phases of User Interface Design

4.1 Phase I: Goal Setting to Functional Model

The design process usually starts with some sort of requirements document, preceded by negotiations amongst various groups in the organization. Requirements are usually expressed as specific features or ways in which the system should behave. Dialogue is necessary in order for the designer to determine the intent behind the requirements as written.

4.1.1 The importance of clearly identified goals

The designer should not attempt the design process without working out a clear and

agreed upon set of objectives with all groups involved. As noted in [22], "A system developed without specific, agreed objectives is rather like the old adage of Christopher Columbus' voyage of discovery: 'That he set out not really knowing where he was going, arrived not really knowing where he was, and came back not really knowing where he'd been.'" If early agreement on objectives cannot be reached, "no level of technical sophistication nor inspired project management can rescue the situation." [14].

The primary goal, in the case in question was to design an organizational messaging system. The designer starts by working out sub-goals, i.e. the implications of an "organizational" system. That is, it must include both administrative and personal messaging features. It requires flexibility in the basic functions in order to support a wide variety of users, and it must satisfy the needs of the organization in terms of administration and control.

Next, the designer lists the basic functions by asking some general questions. One such question is "How should the user perceive the system?". The answer chosen in the messaging design was that the user's working environment in the system should be analogous to his existing mail capabilities in the office and in particular on his desk-top, but only where this makes sense. It is neither useful nor desirable to transfer every existing feature into the electronic environment [22].

4.1.2 Developing the Functional Model

This approach leads to an in-basket analogous to the existing one. The in-list provides the overview that a person obtains by rummaging through the office in-basket and glancing at the cover page of each document in it. Direct transfer of the existing out-basket concept did not seem appropriate since it is mainly a dispatch holding area, and messages are dispatched immediately in the electronic environment. Instead the electronic out-basket was conceived as analogous to the electronic in-basket. The in-list is often used as a "to-do" list. The out-list would be used as a "to-follow-up" list and the out-basket becomes a holding area for messages being tracked.

At this point, the designer is building a model. The model must be tested and if necessary revised, based on participative design and on real use. In the case of a product, participative design techniques are not directly applicable, but in-house or pilot customer prototype systems can be used to provide some of the necessary feedback.

Another question for the designer is "How does the user perceive his organization in the system?". This leads to the concept of an organization coordinator/ombudsman, a shared organization file space, and a directory of all users in that organization (analogous to the office phone directory). The question "What control over usage costs will the organization deem essential?" leads to the concept of the coordinator allocating private file space to users in the same way that an office administrator allocates pencil and paper, or secretarial support.

Another key question "What does it mean to a user to read a message?" leads to a comparison between the standard use of the verb "read" and its meaning in the electronic context. In natural language "read" implies an understanding of content. In a computer activity, it only implies the presentation of information to the user on a screen or paper. One result of this is that if a message system

tells a sender that a message has been 'read', the sender is likely to assume 'read and understood', which may or may not be true [20].

These are only samples of a long list of questions that the designer asks at this stage. The resulting answers and the analogies to the existing environment are important to the design, but are also very useful for discussion with users and in writing user documentation.

4.1.3 Some Guidelines

At this stage, the designer usually investigates other similar systems, in order to find out the issues in provision of the basic functions. For example, existing message systems support different feature sets; have different understandings of the same feature, e.g. the out-basket; or features work in different ways, e.g. in-list resequencing.

It is also important at this stage to review the general principles of human - computer system communication ([2], [6]), and to establish a set of specific guidelines which are particularly relevant to the current design. Those key to the messaging design were:

1. The system is to be individually adjustable,
   and
   it should appear simple to users with simple needs.

2. The user has control if he so desires,
   but
   in some areas the organization requires overall control.

3. Things that are the same should look the same,
   and
   things that are not, should not.

4.1.4 Results

The result of this phase of the design is a concepts paper giving a framework for the system, within which specific features can be defined and can evolve. The designer must be able to prove to the satisfaction of his customer that this framework addresses their very specific requirements and constraints. There are many discussions starting "But what if...?" in which the designer finds that he did not completely understand the requirements, and the customer discovers a technical possibility that enhances the functionality. This results in iteration, and some compromise as well as improvement to the initial design. It is critical that a satisfactory functional model be agreed upon at this stage. It is not only essential to further design, but also to later system evolution.

4.2 Phase II: Scenarios of Use to Dimensions of Use

In order to translate the functional model into a detailed feature set, the designer must understand who the users are. It is very dangerous to assume an average user since there is no average user, and "building a system to satisfy the average user is likely to result in a system that does not really meet any user's needs." [8].

Thus, the next step in the design process requires development of scenarios of use for different classes of users. Ideally this is done with the users themselves in the participative design approach [10]. In the case of product design, users might be represented indirectly by marketing, or designers could have direct contact with interested customers, or the users of a prototype system.

4.2.1 Examples of Scenarios of Use

Some of the scenarios of use identified in the organizational messaging design are summarized below:

- A specially trained clerk works full-time in a teletype room. This person accepts handwritten or typed messages from all over the organization and enters them into the system on behalf of their originators.

- A select group service with a secretary sending and receiving messages on behalf of one or more managers. The secretary may or may not deal directly with message content.

- A manager interacts directly with the system on his own behalf.

For each class of users, the designer must examine each of the basic functions, in order to determine if it is useful for that class, and if so, how it should be adapted to their use.

For example, in the teletype room scenario, it seems most useful to have all messages auto-delivered, and to have the in-basket automatically cleared once delivery is complete. This class of user may not want to be aware of the existence of the electronic in-basket. They are unlikely to use private files.

In the case of the select group service, the secretary may peruse incoming mail and take action on the manager's behalf. The mode of use would vary according to the responsibility delegated by the manager. It is likely that the manager would use the in-list as a "to-do" list and to retain complete control over clearing it (as would the secretary to whom he has delegated complete messaging responsibility). He would probably use private files.

As can be seen from the above descriptions, there is no distinct boundary between different sorts of users, and indeed there is room for a great deal of individual variation in any of these cases. In developing these scenarios, the designer should consider the organizational impact of the system, that is implied changes in the user's job content, responsibility, autonomy, and job satisfaction. Of course, he cannot fully do this, as it is essentially an organizational issue that must be solved within the organization. Participative design or design by prototype will help the designer in developing scenarios of current and anticipated use and in analyzing their impacts.

4.2.2 Determining Dimensions of Use

At this stage, the designer is attempting to map the two-dimensional average user versus basic function view of the system, into the more general one of a variety of users versus features tailored to suit their preferred mode of interaction and organizational perspective.

In order to determine the dimensions of the mapping, it is important to find out what distinguishes the requirements of the different classes of users. The following lists some examples of dimensions of use that emerged during the organizational messaging design process.

a) Direct or Mediated Use:

Use of the system is considered to be direct if a person interacts with it to deal with message content, for example to read a message and compose a reply online. Use is mediated if a person interacts with the system on behalf of someone else. That is, the user is an intermediary who deals with the message but is not responsible for its content. This dimension affects such features as mode of message composition and use of filing.

b) Frequency of Use:

Whether a person is a casual or a full-time user determines to a large extent the degree of experience he has with the system, and therefore how much guidance he should receive.

c) Active or Automatic Mode of Interaction:

Use of features is defined to be active if the user takes responsibility himself for functions, such as deleting messages he no longer needs. Use is automatic if the user has the system do it for him, for example by having it always clear messages from the in-basket under fixed conditions.

Whether a user prefers an active or automatic mode of interaction depends on his job responsibilities as well as personality - some users show a tendency to follow the human principle of least effort. Also full-time users with a high volume of messages might choose an automatic mode to help in coping with information overload.

d) Message Trafffic and Message Flow:

This refers to the number of messages dealt with in a certain time period and whether they are mostly sent, mostly received, or balanced between the two. Automatic delivery can assist users with high incoming message traffic and selection options help a user to find messages in a full in-basket or out-basket.

e) Message Broadcast Patterns:

This is determined by the number of recipients on the address list of a message. In administrative messaging, messages are often sent to the same recipients on a more or less regular basis. Features like pre-defined distribution lists help to support this.

f) Terminal and Line Speed:

Finally, the user community can have a wide set of existing terminal types and different line speeds. These have an impact on the editing capabilities to be provided and on the general form of the interface.

4.2.3 Dimensions of Use Related to Classes of Users

In the examples above, use by the teletype room class would tend to be full-time, mediated and automatic, with high outgoing message traffic. They might have a limited use of the note-file for distribution lists. Messages could be automatically cleared from their in-baskets after auto-delivery.

The secretary class would use the note-file to a greater extent, for example to save the message sheet when asked to compose an urgent message. They could have all read messages automatically cleared from their in-baskets at the end of a session. They might want to receive notifications of new incoming messages during a session. Their use is also mediated.

The manager class are direct users, with a tendency to casual usage and a lower volume of mail. They might be expected to use the note-file extensively, for example to compose a message over a long period of time or to save incoming messages for later action. They would probably want to clear messages from in-baskets and to resequence, by command.

Note that in the first two examples users need not be aware of unused features like the wastebasket, but the feature is still available. This continuity of features between classes means that a user can naturally change his mode of interaction as his use evolves or his sophistication of usage increases.

In terms of the dimensions of use, administrative messaging tends to be mediated, full-time, and with high message traffic. Personal messaging is usually direct, more casual and with lower message traffic.

4.2.4 Results

The result of this phase of the design is a document on models of usage: identification of classes of users and of the dimensions of their use. Classes of users and the scenarios of their use are key for communication with marketing or the steering committee, with system choosers and with the users themselves. Dimensions of use are key to the next stages of design, to design justification and to consensus generation. Both are important for iteration back to verify the functional model

4.3 Phase III: Feature Sets and Their Personalization

Having identified classes of users and dimensions of their use, the designer can now attempt a detailed mapping of their requirements into feature sets, keeping the guiding principles in mind. This along with consideration of interaction style, will determine feature personalization.

At this stage, the designer also defines the type of dialogue as well as the amount of system assistance to the user [5]. The form of the dialogue, e.g. command syntax, is defined here, but the exact vocabulary is usually not finalized until software development is underway, and a full set of exception situations have been identified.

It is also important to consider how much organizational control to build into the system, in order to satisfy both system choosers and users, who naturally have different points of view on this [14]. In order to support both organizational and user control in the messaging system, a user's profile was divided into a set of options and a set of permissions. Only the organization coordinator could modify

permissions, such as allocation of private filing space, and only the user himself could modify options, such as the language of interaction.

The decision as to what should be a permission and what should be an option requires careful thought, consultation with the customers, and some understanding of the psychology and social dynamics of people in an organization.

4.3.1 An Example

The issue of tracking of messages is one controversial example of an area of potential conflict between organizational control and the individual. Tracking of delivery is essential from the organizational point of view, particularly in its administrative messaging applications. However many individuals consider tracking of messages into their electronic environment a serious invasion of privacy.

This led to a specification of tracking such that the sender can tell the system to track delivery of a message, but the recipient has full control of when delivery confirmation occurs. This can be active or automatic. The system informs the recipient that the message is being tracked by use of an in-list flag and also a prompt when the message is read. The recipient can acknowledge delivery by entering a simple command, analogous to signing for a registered letter. Alternatively he can specify the confirmation condition as a profile option for automatic confirmation. He can also choose to ignore the tracking request.

When the sender receives confirmation of delivery, all he knows is that the recipient has been made aware of the existence of his message, and not that he has either read or understood the content. The latter is not under the system's control. The system provides a tool for tracking - how it is used will be influenced by organization policies, but ultimately is under the control of individual users.

4.3.2 Implications of Analysis of User Classes on Personalization

It is important that the analysis of user classes be used to allow the system to be adjustable for individual users and not to enforce specific modes of use. A profile and feature package can be recommended for a particular class of users, but any user can set his own individual options.

When user classes were identified in the messaging design, it became apparent that both automatic clearing of the in-basket, and active clearing by user command, were desirable. This led to options in the user's profile to specify either that the user actively crumples in-basket messages or that the system should clear it automatically under certain circumstances. The wastebasket is available, but unobtrusive in automatic use.

There are a variety of profile options for in-basket and out-basket management (clutter management), resequencing, tracking, and so on. Analysis of classes of use leads naturally to personalization, by identifying how a feature should be provided according to different dimensions of use. There are additional options relating to general interaction style, for example language of interaction and verbosity of system prompts.

We hypothesize that full-time, direct, active, high volume, experienced users will tend to use the most general form of features. If the simpler forms are then de-

fined within this context, usage can evolve naturally. In the example above, a user with the automatic clearing option need not be aware of the wastebasket, how to investigate its contents, and how to uncrumple messages from it. However it is there and he can look into it at any time. At some point, as his mode of usage changes, he may decide to reset his profile option to crumple by command.

4.3.3 Caveats for the Designer

Most of the detailed user interface design falls into this phase. The following are some general caveats for the designer:

1. There is a tendency to avoid user interface decisions by providing an excess of options that confuse the concepts and become a source of inconsistencies.

2. The ongoing design effort is usually underestimated.

3. Difficulties in the design are often overlooked and only seen at the detailed level. Exceptions are to be expected in all design phases.

4. The designer's previous experience will, consciously or subconsciously influence his current work [22]. A design team with different backgrounds, and good review help balance this.

5. The law of routine ignorance [23], that is "the more often we do something, the more we tend to forget how we do it.".

4.3.4 Results

The result of this phase of design is a detailed user interface specification. This is an important review point. During the initial phases, the designer's focus for communication moves slowly from the customer and steering committee to developers and users. All should be included in the review process. This is the point at which development can start, and there will be iteration of the user interface design based on input from developers.

4.4 Phase IV: User Documentation and Finalization of the Interaction Language

The user interface specification is one basis for software design and development (other specifications define operational interfaces, administration and so on). This results in dialogue between the user interface designer and the developers, and iterations on feature design. Once this has stabilized, user documentation can be completed.

4.4.1 User Documentation

The content of user documentation is similar to that of the interface specification, but its perspective and its structure should be very different. Unfortunately, writing user documentation is often considered to be a very minor part of the design and development cycle, and the right resources are not assigned to the task.

Good user documentation requires, in addition to sound writing skills, the ability to understand the point of view of users, and a knowledge of the design concepts and the implementation. The latter is important in order to explain to the users how to deal with error situations and the quirks of the system, that is "what to do if...". In addition, interesting analogies and illustrations help make documentation palatable to the user. The analogies used to develop the functional model can now prove useful in explanation to the user.

Thus, user documentation should be produced as a joint effort between writer, designer, programmer and psychologist. If possible, it should be discussed with users to ensure their understanding. In the product situation, this becomes more difficult, and care must be taken not to bias the documentation towards one class of user.

It is desirable that the documentation team include one person, usually the writer, who is new to the system, in order to introduce a fresh perspective. With familiarity, user interface and software designers become comfortable with their own jargon and can lose track of what will be hard for the user to understand [22].

User documentation is very often poorly structured and forces a user to read an entire document in order to gain an understanding of the 10% of features, in which he is initially interested. Structure becomes even more important when the documentation must communicate with a variety of classes of users, with different dimensions of use. User documentation can be prefaced by a short list of questions, aimed at identifying dimensions of use. The answers are then linked to a recommended "reading list" of sections, needed by that person to start using the system. The intent behind this is to present a system with inherent flexibility, while minimizing complexity for an individual user.

Online documentation or help should remind the user of what he has read in a user guide, using the same style and terminology. In general, online documentation should be short and succinct (although extensive computer-aided learning modules might be used as a training tool). A maximum of a screenful of information, or a few lines on a teletype, seems reasonable for a help request from the user. A user frequently requests help when he is in doubt about a specific thing, for example the exact syntax of a command, or a particular command verb. To find out how to perform a sequence of actions, he would be more likely to consult the user guide, or his ombudsman.

4.4.2 Finalization of the interaction language

When the user guide is near completion, and before system implementation is finished, it is important to fix the language of interaction between user and system. The language syntax must be defined before development can start, but the exact wording of commands and the language and style of system messages, menu or help information can be finalized later. Guidelines on this and common pitfalls are well documented in the literature ([24], [20]).

4.5 Evolution

Design and development is iterative, and changes will occur after documentation is complete. It is very important that user documentation, epecially online documentation, be kept up-to-date during system evolution.

Implementation into the users' working environment will not be covered here, but the user interface designer should be involved at that time, both to explain the intent of the concepts and features and to learn by feedback from implementation and usage problems [8]. This will lead to a better understanding of the users and their requirements. Participative re-design with users can then occur on the basis of initial use, and this will be key to further iteration of the design.

The functional model is very important in system evolution. Design integrity must be monitored. The ad hoc addition of extra features without consideration of the model and classes of users can undo a good user interface, just as it can destroy a good system design.

The dimensions of use can be valuable in the evaluation of feature extensions and their justification. As stated in [14], "When criteria for success are ambiguous and lack a simple measure, decisions concerning system extensions will appear arbitrary from the perspective of some of the interested parties."

In addition to monitoring design integrity, it is also important to continually evaluate its applicability. Use evolves over time [25], and the dynamics of an individual's job must be reflected in a system whose intent is to make the person more effective in doing that job [14]. Thus we postulate that if a design is useful, it will soon have to be changed.

5. CONCLUSION

In a comparison of software design to engineering design, Malhotra et al. state that it is "very difficult, in fact, to ascertain whether the design is complete, is consistent, or whether it meets the functional requirements. This seems equivalent to not being able to tell if a building will stand up!" [13]. The user-oriented design approach described, and in particular the dimensions of use, can help in justification of design.

Identification of dimensions of use can be carried much further for various types of office products and value-added services. Continuation of existing research into understanding office procedures ([26], [27], [28]) can provide insight into scenarios of use, that lead to dimensions of use. This research also provides a base of understanding that is important to all phases of user-oriented design, and key to implementation. The approach described here can also be applied to the design of administrative user interfaces to a system. This area is very important to successful system implementation, but usually receives much less attention than the general user interface.

User interface design is a discipline that requires special knowledge and skills, just as compiler writing, data base design, or architecture do. A designer or design team should have, in addition to a grounding in computer science: some basics of user interface specific knowledge from psychology, such as on response time or memory span; from linguistics an understanding of the syntactic and semantic principles of language; and some background in sociology and organization theory, relevant to issues of job design and organizational control.

Currently, good user-oriented designers are rare. There is a need to educate computer scientists in the skills and knowledge outlined above. Psychologists and linguists with a good understanding of computer science are also a rare breed, and there is a need to educate them also. Since it may be difficult to find these skills in one individual, the designer must be a good teamworker. Guidelines exist

already for design of some more or less standard user interfaces, such as data entry or reservation systems ([1], [5]). However, the designer who deals with a new sort of system must develop a new approach.

REFERENCES

[1] J. Martin; Design of man - computer dialogues; Englewood Cliffs, NJ,1973.

[2] W. Dehning, H. Essig, S. Maass; The adaptation of virtual man - computer interfaces to user requirements in dialogue; Report of the Department of Informatics, University of Hamburg, W-Germany, July 1980.

[3] B. Shneiderman; Human factors in designing interactive systems; Computer, December 1979.

[4] J. Gilb, G.M. Weinberg; Humanized input - techniques for reliable keyed input; Cambridge, MA, 1977.

[5] K.D. Eason; Dialogue design implications of task allocation between man and computer; Ergonomics, vol. 23, no. 9, September 1980.

[6] B. Shackel (ed.); Man - computer communication, vol. 1, summary report; Infotech State of the Art Report, Maidenhead, England, 1979.

[7] S.L. Smith; Requirements definition and design guidelines for the man - machine interface in C3 system aquisition; Mitre report no. M80-10, April 1980.

[8] M.J. Ginzberg; Steps towards more effective implementation of MS and MIS; Interfaces, vol. 8, no. 3, May 1978.

[9] F.G. Withington; Coping with computer proliferation; Harvard Business Review, vol. 58, May-June 1980.

[10] E. Mumford, F. Land, J. Hawgood; A participative approach to the design of computer systems; Impact of Science on Society, vol. 28, 1978.

[11] J. Bair; Avoiding working non-solutions to office communications system design; Proc. of COMPCON Spring 1980.

[12] P.G.W. Keen; Information systems and organizational change; Communications of the ACM, vol. 24, no. 1, January 1981.

[13] A. Malhotra, J.C. Thomas, J.M. Carroll, L.A. Miller; Cognitive processes in design; International Journal of Man-Machine Studies, vol. 12, 1980.

[14] P.G.W Keen, E.M. Gerson; The politics of software design; Datamation, November 1977.

[15] J. Vittal; MSG - a simple message system; Proc. of the International Symposium on Computer Message Systems; Ottawa, April 1981.

[16] T.H. Myer, C.D. Mooers; HERMES message system user's guide; Bolt Beranek & Newman, Boston, January 1977.

[17] J. Vallee, H.M. Lipinski, R.H. Miller; Group communication through computers, vol. 1 (Design and use of the FORUM system); Institute for the Future report no. R-32, Menlo Park, CA, 1974.

[18] M. Turoff, S.R. Hiltz; Development and field testing of an electronic information exchange system: final report on the EIES development project; NJIT Computerized Conferencing and Communications Research report no. 9, 1978.

[19] S. Rohlfs; User interface requirements, Infotech State of the Art Report on 'Convergence: Computers, Communication, and Office Automation', Maidenhead, England, 1979.

[20] S. Rohlfs, Linguistic considerations for user interface design, Proc. of the Workshop on Integrated Office Systems, Rocquencourt, France, November 1979.

[21] G. Millard, H. Williamson; How people react to computer conferencing; Telesis (3), 1976.

[22] K.R. London; The people side of systems; New York, 1976.

[23] A.H. Konstam; The user - designer conflict and its resolution; Proc. of the 7th Annual Computer Personnel Research Conference, Miami, 1980.

[24] T. Stewart; Communicating with dialogues; Ergonomics, vol. 23, no. 9, September 1980.

[25] S.R. Hiltz, M. Turoff; The evolution of user behaviour in computerized communication systems; Proc. of the International Communication Association Conference, Acapulco, Mexico, 1980.

[26] C.A. Ellis, G.J. Nutt; Office information systems and computer science; Computing Surveys, vol. 12, no. 1, 1980.

[27] L.A. Suchman; Office procedures as practical action: organization theory and system design; Proc. of the 5th International Conference on Computer Communication, Atlanta, Georgia, October 1980.

[28] E.H. Wynn; Computer message systems as a unique medium of communication; Proc. of the 5th International Conference on Computer Comunication, Atlanta, Georgia, October 1980.

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP,1981

ACCEPTANCE RESEARCH STRATEGIES IN COMPUTER MESSAGE SYSTEMS

Reinhard Helmreich

Labor fuer Benutzerforschung
Zentralbereich Technik, Siemens AG
Muenchen-Perlach
Federal Republic of Germany

1. INTRODUCTION

Communication is a fundamental form of human behavior. It is in this light that the desire to place new technical media at the service of human communication is to be seen. The computer is regarded as the ideal tool with which this communication can be improved.

This does not only mean overcoming the barriers of space and time separating the communication partners; nor is it simply a matter of speed. It is rather the expectation of qualitative improvements from new communication media. Thus it is hoped that the computer will, for example, lead to an increased transparency of decision processes and cooperation on a more equal footing between group members.

In contrast to these great expectations, experience has shown that new developments in the field of communication engineering are not always well received by the user. New possibilities will be rejected, despite - when they are considered in isolation - a high degree of technical maturity.

The computer as a communication medium is therefore not merely to be viewed from a technological standpoint. Sociological and psychological considerations must also be taken into account, in both the planning and implementation of systems investigating the phenomenon of "acceptance". Acceptance has thus become a subject for research, the object of which is to avoid bad technical developments which fail to satisfy the needs of the user; this also includes preventing any undesirable consequences from arising in the economic and social spheres.

This involves factors, into whose interactions little research has been done. They concern:

- device-specific characteristics
  (e.g. range of functions, technical maturity, availability, command language, ergonomics)

- application-specific characteristics
  (e.g. quality and quantity of the workload, organization of the work, structure of the tasks)

- user-specific characteristics
  (e.g. ability and previous experience, initial attitudes, motivation)

Acceptance research can thus be seen to be an interdisziplinary exercise which incorporates technical, organizational and soziopsychological methods.

In West Germany the need for acceptance research is recognized by the Federal Ministry for Research, which is increasingly linking the promotion of technical developments with a parallel social science program.

In the following description of acceptance research strategies two rival approaches are compared (Section 2). Section 3 contains a report on experiences with these research strategies in a field test on the acceptance of computer-supported communication.

2. RESEARCH STRATEGIES

The question of the acceptance of technical media was for a long time almost exclusively studied using empiric-analytical techniques. Now, however, there is a tendency to change the scientific method. The "other" strategy is called 'action research'.

First, it should be said that the discussion about method has undergone a polarization as a result of the two competing techniques, the empiric-analytical method and action research. There are THE empirics on the one hand and THE action researchers on the other. The two groups represent apparently irreconcilable positions.

There now follows a pointed comparison in which the two approaches are briefly characterized (Sections 2.1 and 2.2). Next, an attempt is made to bring the two extremes together - for the investigation of the acceptance of technical innovations in the communication field it is necessary to develop a research strategy which combines the traditional empiric-analytical approach with action research (Section 2.3). Finally, an account is given of experiences with such a strategy (Section 3).

2.1 CHARACTERISTICS OF THE EMPIRIC-ANALYTICAL APPROACH

This research strategy is based on the criteria of a traditional scientific approach. Data on all seemingly relevant characteristics are collected by observation, questionnaires and interviews. The evaluation of this data is carried out according to statistical requirements and finally they are interpreted through the eyes of the researcher and his employer. The aim is to obtain, for the most part, comprehensive and repeatable results.

The relationship between the researcher and the people involved in the study is characterized by distance: the researcher must not be allowed, by his research, to affect the characteristics to be measured.

If clear hypotheses about relationships are to be formed and actual states which can be defined in advance are to be included, then the empiric-analytical research strategy is the appropriate method.

2.2 CHARACTERISTICS OF ACTION RESEARCH

In this research strategy the researcher himself goes out into the field to be investigated.

The aim of this research is not to record as objectively as possible phenomena which have been defined in advance. Instead, action research is concerned with producing and testing changes in the field being investigated. This must be carried out by cooperation with the users of the new technology, both parties having equal status. The functions of the system of technology are adapted in an iterative process to the expectations of the user. The researcher's job is thus to convert his specialized knowledge into direct action and the result of the research is to ensure the successful introduction of the new technology.

2.3 PROPOSAL FOR A COMBINED STRATEGY

The first application of new communication technology for research purposes usually progresses according to the following sequence of steps:

- First, a field of application has to be found and analyzed for apparently relevant characteristics
- Those involved and their environment must be prepared for the the new technology
- A "change agent" promotes the necessary changes in the introductory phase
- The "change agent" also ensures the feedback on experiences, in order that technological improvements can be made
- Quite independently of this, an evaluation of the use of the equipment must be carried out which should be as objective as possible

Thus it is reasonable to use both action research and the empiric-analytical research strategy in studies on the implementation of new communication technology.

From the action research point of view, users and researchers must jointly plan and prepare the use of equipment, exchange experiences, discuss problems and overcome difficulties.

From the empiric-analytical point of view, the conditions for acceptance must be recorded and evaluated. It must be possible to derive the reasons for acceptance or non-acceptance from the characteristics recorded.

Hence there are two very different tasks facing the researchers concerned. On the one hand, they place their specialized knowledge at the service of the implementation of the new technology. They discuss and evaluate observations and documents during the course of their work. They are the recipients of feedback, hold discussions with both individuals and groups of people and, using

suitable technical means, intervene to give impetus to the learning process of the user. The main result of the research is to ensure a successful implementation of the new technology. As a secondary consequence there is a documentation of the work in the form of a research report.

The other task facing the researcher is the application of suitable techniques for the measurement of characteristics which have been recognized as important and the subsequent statistical evaluation. The result of this research is a report on the conditions and effects of defined characteristics.

Such a combined strategy is on the one hand an advantage in the achievement of a successful implementation by the user and on the other makes generalizable data available, with the aid of which conditions and effects of acceptance can be explained.

3. EXPERIENCES

A computer message system is currently the subject of a broadly-based field study being carried out by the Central Division Technology of Siemens AG, Munich. This computer message system is part of a functionally powerful, integrated office system for the support of managerial tasks.

It must be emphasised that the questions which are to be answered during the course of the field study are not unrelated. Indeed, it is the interaction of the various factors which is the main object of our interest:

- How are system implementation and user training progressing?
- Is the full potential of the system being used?
- What capabilities are required of a computer system if it is to perform satisfactorily under normal working conditions?
- What factors encourage or inhibit user acceptance of the new technology? What are the consequent effects?
- What demands are placed on the system user?
- How user-friendly is the system?

The system was initially installed in January 1979. The participants were not preselected. Any initiative to increase the size of the group had to come from the users themselves. In this way the number of users rose to over 60 within the first two years of the field study.

Equal weight is given to the application of the empiric-analytical method and action research, and a separate research team was assigned to work independently towards the respective objectives. This division of labor is designed to prevent any mutual influence or mixing of the research objectives.

The action researchers have, together with the users, successfully introduced and continually improved the computer message system. The research team following the empiric-analytical method has been able to collect an enormous quantity of data relating to the potential use of the new technology, the communication structure and communication activities of the users with respect to task-specific acceptance factors and those specific to people.

This now opens up the possibility for the method employed by the action researchers to be evaluated and so objectified by means of the empiric-analytical methods.

Neither research strategy can, however, be completely isolated from the other. Compromises have to be made, particularly in the case of the empiric-analytical method. Despite a strict division of personnel in the two research groups interference arises in the following areas:

- Action research leads to an enlarged group of participants. Hence, it does not permit the formation of a "representative" random sample which the empiric-analytical strategy would in fact require for the specification of the statistical evaluation process and justification of genenralizable data.

- On account of the constant intervention, action research continually creates new conditions in the field under investigation. This makes the planning of instrument application in the empiric-analytical methods more difficult. The repeated application of measuring instruments is to some extent rendered impossible because the quality of the conditions has changed so much that the instruments can no longer be used.

- The action research method gives those involded a feeling of being constantly attended to. This produces a "Hawthorne effect", which is useful for the introductory phase of the computer message system on the one hand, but also makes the technology-dependent acceptance factors hard to identify on the other. The validity of the empirical data is thereby diminished.

Experiences with new technical media should not merely be chance observations. In fact acceptance research is indispensible; it is a field which must be developed both in content and method. This contribution aimed to show how sociological strategies could be comined with recent action research approaches.

The great expense involved on the method side of acceptance research is doubtless justified in view of the great technological expense of computer message systems. The expectations placed in the results of acceptance research must not, however, be overestimated. The empiric-analytical method can only make statements about actual states and reveal weak spots in existing systems. Action research can only work on systems which, in principle at least, are operational.

Even the most intensive acceptance research can be no substitute for the creative achievement which must be the basis of every technological development. This research can only indicate the direction further development should take. It also provides facts which permit the effects of new communication systems to be estimated.

REFERENCES

[1] Becker, H., Organisationsentwicklung, Z. f. Arbeitswiss. 31, (1977) 203-208

[2] Clark, P.A., Action Research and organisational change. London: Harper & Row (1972)

[3] Helmreich, R., Was ist Akzeptanzforschung?, Elektronische Rechenanlagen 22, 1 (1980) 21-24

[4] Hertog, F.J.d., Arbeitsstrukturierung: Experimente aus Holland. Bern: Huber (1978)

[5] Jenks, R., An action research approach to organizational change. J. Appl. Behav. Sc., 6 (1970) 131-150

[6] Kubicek, H. Humanisierung des DV-gestuetzten Bueros durch partizipative Systemgestaltung. Angew. Informatik 20, (1978) 131-150

[7] Moser, H., Praxis der Aktionsforschung. Muenchen: Koebel (1977)

[8] Mumford, E., The participative design of clerical information Systems. Paper for the Copenhagen Conference on Computer Impact-78, October 25-27 (1978)

[9] Rogers, E.M. & Shoemaker, F.L., Communication of Innovation. New York: The Free Press (1971)

[10] Schneider, U., Sozialwissenschaftliche Methodenkrise und Handlungsforschung. Frankfurt: Campus (1980)

[11] Stiegler, B., Kooperationsformen bei arbeitnehmer-orientierter Aktionsforschung. Z. f. Arbeitswiss., 32 (1977) 129-134

COMPUTER MESSAGE SYSTEM
R.P. Uhlig (editor)
North-Holland Publishing Company
© IFIP, 1981

HUMAN FACTORS REQUIREMENTS FOR MANAGERIAL USE OF COMPUTER MESSAGE SYSTEMS

Julian Newman

Department of Information Technology
City of London Polytechnic
London, United Kingdom

Human Factors recommendations for Management-oriented Computer Message Systems must be securely based in an analysis of the managerial tasks it is to support. The paper considers the Human Factors implications of existing knowledge about the managerial task, and some factors which are currently affecting the functions of management. It is argued that augmenting the effectiveness of knowledge-work, and its integration into the decision-process should be a central design concept for management-oriented Computer-based Message Systems. This leads to a view of interface design that emphasises information visualisation, decision-support and feedback through computer conferencing. Further implications of the approach are briefly discussed.

HUMAN FACTORS AND THE MANAGERIAL TASK

This paper is concerned with the Human Factors requirements for those computer-based message systems which are primarily designed to promote the effectiveness of management; it is not therefore concerned with those issues that are general to all interactive systems. The systems envisaged must support the interaction of line, staff and top management, facilitating both the internal integration of the organization and its environmental responsiveness. Recommendations concerning the Human Factors requirements of a management-oriented computer message system must be securely based in an analysis of the managerial tasks it is to support. Rapid changes in the environment of management pose a challenge for the Human Factors analyst; this paper attempts to synthesise a practicable design-perspective from a consideration of management theory, organization theory, and behaviourial studies of man-computer interaction.

The importance of task analysis for the Human Factors design of information systems has been well argued by Tom Stewart (1976a):

> The fundamental requirement for formatting displays is that the communication from the computer should be organised and structured in a form appropriate to the task and frame of reference to the user. Where several users may wish to perform several tasks with essentially the same information content, it may be necessary to compromise any single user-task requirement in the interests of economy. It should however be pointed out that the way in which information is organised and presented may determine the way in which the material is used or the conclusions that are drawn from it. Simply providing users with the 'right' information and

> disregarding its presentation may be a false economy. Badly or inappropriately formatted displays waste time and cause errors to be made both in reading and in interpretation.

Hammer and Zisman (1980) have emphasised that successful office automation depends upon identifying and making more effective the office's business function, rather than merely automating existing procedures. Rosemary Stewart (1973) has suggested that the impact of computer systems on management has been limited by systems designers underestimating the complexity of business, overestimating the importance of formal information to management, and neglecting Human Factors in work design. Computer message systems may be conceived in 3 different ways:

1 Electronic mail
2 Electronic meetings
3 Knowledge-work augmentation systems

Each of these approaches will yield different task-fit characteristics. Systems of any kind will not be used unless they are well suited to the tasks of the intended user. Interactive computer systems, especially, need to be used regularly if their benefits are to be realised. Poor fit between a manager's tasks and the informatic tools provided generally results either in remote use through an intermediary (thus losing opportunities for familiarisation, learning and growth), or partial use (failing to use the power of the system effectively) or even total disuse (Eason, 1976b). The effectiveness of a message service is strongly dependent on ease of access to the system, on how often people look at their messages and on the capabilities the system offers outside pure message processing. For example, Stotz et al (1979) described a situation where users found high task fit because the system (SIGMA) was potentially much more than a message system. A fundamental question about management-oriented Computer-based Message Systems is this: do the additional facilities provided in the "augmentation systems" approach, over and above those found in Electronic Mail and Computer Conferencing systems, improve the task fit sufficiently to justify the greater resources required for their adequate implementation? This can only be answered by an analysis both of the current pattern of management activities, and of the environmental pressures that are shaping the future management-structure of organizations.

Patterns of Managerial Work

Several studies of managers' work patterns have been carried out by R Stewart (1970, 1976). Methods used included self-recording by managers, direct observation, and intensive study of a smaller sample. One study (R Stewart, 1970) showed that on average 36% of a manager's time was spent on paperwork, while only 6% was spent telephoning (however other studies - eg Horne & Lupton 1965, Copeman, Luijk & de Hanika, 1963 - suggest that a figure of 10 to 15% would be more typical). The intensive study (R Stewart, 1976) included an analysis of mail contacts; the large majority of mail received was internal, was of a routine nature, and for more senior managers its handling was largely delegated. Most of the outgoing mail was also internal to the organization. The manager with the largest mail sent out an average of 24 letters/memos per week. 50% of these went to his subordinates. Only 33% were responses to mail received. Stewart concludes: "The general picture from the mail records and from our other information is that mail was a minor, mainly routine, aspect of their jobs."

Mintzberg's study of chief executives emphasises their aversion towards this medium of communication. Fear of being overwhelmed by a growing backlog, and preference for live action, led them to treat most mail items in a rapid and cursory fashion (Mintzberg, 1973).

There are of course important differences amongst managers in the way their time is budgeted. Stewart used a cluster-analysis program to classify managers on the basis of their self-recorded use of time. One group, about one fifth of the sample, spent nearly 60% of their time in paperwork; but even they spent 33% of their time in discussions. All groups spent more than 25% of their time on paperwork, on average. Discussions occupied 50% of the average manager's time; however, only a small proportion of these discussions took place in committee meetings. Even the relatively small group whom Stewart identified as "committee men" spent no more than 20% of their time actually in committees.

We may conclude, therefore, that computer message systems will have relatively little impact if they (1) are designed primarily as "electronic meeting" systems, or (2) are designed simply as "electronic mail". If managers are to find it worthwhile to devote the time and effort necessary to master these systems, they will expect and require support for a large proportion of their paperwork activities and probably for many of the functions currently performed by informal face-to-face discussions.

Over and above these considerations drawn from studies of current management practices, we must consider the needs of management, particularly at the highest levels, for data reduction (Ehrenberg, 1971). Emery (1977) identifies one of the major contributors to the complexity of the modern management task as "information overload arising from the communications revolution". More precisely, one could say that disturbances arise not from the communications revolution per se, but for the tendency for the technology for generating messages to run ahead of the technology required for filtering them (cf. Newman & Ward, 1980, Ward & Newman, 1980).

Environmental Uncertainty and the Functions of Management

The nature of environmental uncertainty for the modern organization is radically different from that which obtained a century ago. Organizations consequently need to operate ever more complex models of their environment. In previous papers (Newman & Ward op.cit.; Ward & Newman, op.cit.) we have explored in some detail the characteristics of this uncertainty, the impact of turbulent field effects upon organizations selling in high-technology markets, and the role of knowledge-work in organizational adaptation.

The Systems-theoretic approach to organizations provides some valuable insights concerning changing managerial functions and the organizational environment (cf. Emery (1977), Emery and Trist (1965), Terreberry (1968), Ward & Newman, op.cit.): since economic adaptation in the advanced societies increasingly depends upon the management of information, rather than the management of things, it appears probable that the central design concepts for successful advanced electronic office systems will be (i) communication-oriented decision-support and (ii) augmenting the effectiveness of environment-oriented knowledge-work.

The importance of knowledge-work as a category of economic activity has been demonstrated by Machlup (1962), Drucker (1969), and Galbraith (1974). By knowledge-work we mean the systematic organi-

sation and application of information and concepts.

> Even 50 years ago the market worked well enough for many companies, to the extent that their best strategy for survival and growth was to concentrate on the efficiency of their internal operations .....
>
> The environment of most modern organizations..... poses problems of a qualitatively different kind from those of the classical market. The area under management's control is only a small part of the totality of factors affecting the organization.....

Systems Theory shows why organizations will tend to respond to environmental turbulence by internal differentiation and proliferation of knowledge work functions:

> The law of requisite variety (Ashby, 1956) shows that a system cannot map onto another system for control unless it has equivalent variety within itself. For an organization faced with adapting to a turbulent environment, the requisite increase in internal variety requires the development of appropriate resources for capturing, assessing and applying environmentally-relevant information. Information destroys uncertainty, and information is the stock-in-trade of the knowledge-worker. In post-industrial society organizations are threatened with collapse (Argenti, 1976) not merely if they ignore information coming from the environment, as in the market model, nor only if they interpret environmental information incorrectly, but also if they fail to seek out information of a pertinent nature; for market signals alone simply do not have the channel-capacity to transmit the information needed to describe the environment with any degree of adequacy.
>
> (Ward & Newman, op.cit.)

Managers are involved in knowledge-work in one of two ways, either by having direct responsibilities for knowledge-work (typically but not necessarily in "Staff" functions) or by having access to reports, digests, opinions, etc. produced by knowledge-work specialists. The "augmentation systems" approach, as pioneered by Engelbart (Engelbart, Watson & Norton, 1973; Engelbart, 1975; Engelbart, 1979) aims to support the effectiveness of the organizational knowledge-worker, and to facilitate his integration into the organization's decision-making process.

MANAGERIAL ROLES

Line Management

Many writers have commented upon the unpredictability and variety of the line-manager's role (cf. Eason, 1976). Unlike many staff managers, the line manager generally does not have a body of established knowledge on which to base his practice. The unpredictable nature of his task results from his authority being positional rather than professional. Yet, as indicated above, he does have an ever-increasing need to draw upon the professional knowledge of the staff functions.

> Since it is so difficult to specify the managerial task, it is an immense challenge to specify the kind of management information system that will continue to have high task fit for a manager in a turbulent environment. By comparison with other computer users, managers expect a greater return for a smaller investment of personal effort in using the system, and if this is not forthcoming they will either not use it, or do so in ways which to the system designer appear irrational. This tendency arises mainly from the diffuseness of the manager's role, which makes him reluctant to become too closely involved in specific task contexts.
>
> Newman (1976)

As Eason (1976a) has shown, managers given the facility for extracting information in the form that they need to use it demonstrate a significantly higher "task fit". But the need to give this degree of flexibility and control to a user who also requires maximum ease of use poses major problems of interface design.

Staff Managers and Specialists

The MICA survey (Eason, 1976a,b; Eason, Stewart & Damodaran, 1974; Stewart, 1976b) identified a category of users whom they termed "specialists", whom Stewart (op.cit.) describes as follows:

> The increasing complexity and sophistication of commerce and industry has generated many problems which require detailed and extensive knowledge of specific techniques and disciplines for their solution. In order to solve such problems, industry now employs a wide range of different specialists from financial planners and management scientists to engineering designers and stress analysts....

While Eason and Stewart point up the contrast between these "specialists" and "managers", clearly many of the "specialists" were doing management jobs; the contrast would perhaps have been more happily phrased as between specialists, including staff management on the one hand, and line management on the other. Two important attributes characterise the "specialists" as a group: they need to simplify, quantify and manipulate complex data in order to solve their problems, and they operate within one particular discipline or problem area, thus having problems "filtered" for them to a much greater extent than line or top management. Staff management roles are of course highly differentiated from one another, and while satisfactory task performance implies a high degree of requisite variety in the specialist's occupational behaviour, the routine element in the specialist's work should also not be underestimated (Newman, 1976; Stewart, 1976b). The situation of line managers is the reverse: their roles are less clearly differentiated than those of specialists, but have incomparably greater internal variability.

Task Factors affecting use of Informal Information

Differences in management tasks also affect the relative importance of formal and informal information. In general the tasks of top management are less structured, more "craft-like", than those of middle or junior management. "The more senior the level of management, the greater becomes the importance of informal information." (R. Stewart, 1973). Within staff management there is considerable variation in the degree to which problems are amenable to formal

information-handling techniques. Macintosh and Daft (1978), building on the ideas of Perrow (1967), have shown that where the task is not well understood managers require less information than when there is a formalised body of professional-technical knowledge; thus a diffuse information system is needed for strategic planning or R&D, but an elaborate information system is more suitable for accounting or engineering. They caution against giving specific, detailed information in situations of high uncertainty:

> ... the highly uncertain situations are the ones where specific, detailed information will tend to be inaccurate; to rely on it would be a mistake. In these situations managers should rely on personal judgements, their own and others', and on informal information, to reach a good decision.
>
> Macintosh & Daft (op.cit.)

From the viewpoint of top management, the capability of augmentation systems to give access to both informal and formal information through a common and interactive interface will undoubtedly be a prime advantage of Computer-based Message Systems.

MESSAGE AND AUGMENTATION SYSTEMS AS A MANAGEMENT TOOL

The Nature of Information

As a management tool, a computer message system must be judged by the extent to which it improves decision-making; in other words, by the extent to which it enables the organization to reduce uncertainty by operating an improved model of its environment. Not every message is information:

> The design of information systems ..... demands an understanding of the nature of information, which we define as "input to a decision"; unless a decision is being made, we are dealing with "data" not "information". This distinction is concomitant with the classical definition of information as "that which reduces uncertainty". The answer to the question "what kind of decision is to be made?" determines the kind of information required, and therefore the data needed for that information and the form that the presentation should take. What the decision-maker needs is the ability to manipulate a continuously-updated database, to extrapolate trends in a variety of postulated situations, to visualise the information in appropriate formats, to ensure the simultaneous implementation of the decision over a geographically dispersed area, and the rapid feedback of the decision-consequences.
>
> Newman & Ward (op.cit.)

Manipulating the Database

The augmentation systems approach offers the manager access to specialised software and databases online via a consistent interface, (i.e. one does not have to learn the "quirks" of each and every different computer program or operating system, but can interact with them all in essentially the same way). (Engelbert, 1979). To this must be added facilities for "massage" or manipulation of

numeric data, including:

a) easy interchange between displays of precise detail and of the overall trends.
b) displays for scanning or "browsing" through results, with additional facilities for "homing in" on specific details.
c) easy manipulation of the presentation of the output.

(cf T Stewart, 1976b).

Work station hardware characteristics for top management should include capacity to independently and rapidly update separate windows of a multi-window display, facilities for group viewing of a common screen (eg videobeam displays) and ease of cursor manipulation, by mouse, joystick or trackball. But the message system must also be usable throughout the organization or beyond, and so from less sophisticated workstations also.

Visualising the Information

Information may be visualised either in alphanumeric or in graphical form. Rohlfs (1979) stresses the value of graphics as an aid to cognition: "Complex dependencies between variables are easier to understand when represented as two or three-dimensional graphics. They appear as one 'information unit' to the eye, whereas a table appears as several units which have to be mentally transformed ..... When complicated graphics have to be frequently represented, the use of colour displays should be considered (to enable) the eyes to distinguish between lines....."

However, numerical information is not necessarily best presented in graphic from (Ehrenberg, 1977). Where the task requires "browsing" through data, trying to discern a pattern, as in many staff management tasks, good tabular presentation is often preferable; graphical presentation may fix attention of some kinds of relationships, to the neglect of others. For "browsing" applications, graphic displays often have the disadvantage that the scale of the axes may be adjusted dynamically by the program, making it subjectively hard to compare two successive plots. Of course the user can be given control over the scaling, but this means one more potential distraction from his problem.

Ehrenberg (op.cit.) suggests two criteria for good tabular presentation:

> The Strong Criterion for a Good Table: The patterns and exceptions should be obvious at a glance.
>
> The Weak Criterion: The patterns and exceptions in a table should be obvious at a glance once one has been told what they are.

Ehrenberg adds that a table satisfying the strong criterion will contain some redundancy.

Appropriateness in tables will be task-dependent. If it is a matter of retrieving a single numerical value, then a tree-like "one-dimensional" table is often preferable to a traditional format. One-dimensional tables are less error-prone (Wright, 1977) and quicker to use (Newman, 1978). Table lookup by the user may appear inappropriate in the context of an interactive computer system; but users seeking a single item of information frequently do retrieve a whole

Table 1

Adults who "Really Like to Watch": Correlations to 4 Decimal Places
(Programmes Ordered Alphabetically within Channel)

ITV	PrB	1.0000	0.1064	0.0653	0.5054	0.4741	0.0915	0.4732	0.1681	0.3091	0.1242
"	ThW	0.1064	1.0000	0.2701	0.1424	0.1321	0.1885	0.0815	0.3520	0.0637	0.3946
"	Tod	0.0653	0.2701	1.0000	0.0926	0.0704	0.1546	0.0392	0.2004	0.0512	0.2437
"	Wos	0.5054	0.1474	0.0926	1.0000	0.6217	0.0785	0.5806	0.1867	0.2963	0.1403
BBC	GrS	0.4741	0.1321	0.0704	0.6217	1.0000	0.0849	0.5932	0.1813	0.3412	0.1420
"	LnU	0.0915	0.0915	0.1546	0.0785	0.0849	1.0000	0.0487	0.1973	0.0969	0.2661
"	MoD	0.4732	0.0815	0.0392	0.5806	0.5932	0.0487	1.0000	0.1314	0.3267	0.1221
"	Pan	0.1681	0.3520	0.2004	0.1867	0.1813	0.1973	0.1314	1.0000	0.1469	0.5237
"	RgS	0.3091	0.0637	0.0512	0.2963	0.3412	0.0969	0.3261	0.1469	1.0000	0.1212
"	24H	0.1242	0.3946	0.2432	0.1403	0.1420	0.2661	0.1211	0.5237	0.1212	1.0000

Table 2

The Correlations for the 10 Programmes Rounded and Re-ordered

Programmes		WoS	MoD	GrS	PrB	RgS	24H	Pan	ThW	Tod	Lnu
World of Sport	ITV		.6	.6	.5	.3	.1	.2	.1	.1	.1
Match of the Day	BBC	.6		.6	.5	.3	.1	.1	.1	0	0
Grandstand	BBC	.6	.6		.5	.3	.1	.2	.1	.1	.1
Prof. Boxing	ITV	.5	.5	.5		.3	.1	.2	.1	.1	.1
Rugby Special	BBC	.3	.3	.3	.3		.1	.1	.1	.1	.1
24 Hours	BBC	.1	.1	.1	.1	.1		.5	.4	.2	.2
Panorama	BBC	.2	.1	.2	.2	.1	.5		.4	.2	.2
This Week	ITV	.1	.1	.1	.1	.1	.4	.4		.3	.2
Today	ITV	.1	0	.1	.1	.1	.2	.2	.3		.2
Line-Up	BBC	.1	0	.1	.1	.1	.2	.2	.2	.2	

SOURCE: Ehrenberg, 1977

class of data and then scan visually for the item required, apparently to avoid the "psychological cost" of thinking about how to specify the one item they really need. If they are going to do this, then they should have a facility to reformat their tables in an easy-to-search one-dimensional format (for example by means of a single soft function key). This format decision must be under direct control of the end user, because only he knows whether at a particular time he is using a table to search for a particular value, or to compare values.

Where numerical data have to be compared, traditional table formats will be preferable. But a table often needs a good deal of editing before underlying patterns will emerge clearly. Consider, for example, the level of rounding of tabulated numbers. Too high a degree of accuracy (insufficient rounding) can actually obscure the patterns in the data. Ehrenberg (op.cit.), on the basis of Simon's (1969) discussion of short-term memory, recommends that one should always round to at most two significant or effective digits, since one cannot simultaneously remember additional digits and think about the interpretation of the data. Depending upon the volume of data in the table, the scope for reordering or grouping the data, etc., rounding even to one digit may have advantages. Comparison of Tables 1 and 2 (from Ehrenberg's paper) well illustrates the communicative advantages of rounding and grouping. Ehrenberg offers six basic rules for table design, discusses their psychological bases, and rebuts various objections to their implementation. Ehrenberg's rules are:

1 Round to two significant or effective digits.
2 Use row and/or column averages to provide "visual focus".
3 Figures are easier to compare in columns than in rows.
4 Ordering rows and/or columns by size can bring order out of chaos.
5 Use spacing to guide the eye: excessive spacing makes tables hard to read; figures that have to be compared should be close together, but selective gaps may be used (cf Table 2) to guide the eye as to grouping, etc. in the data.
6 Graphs are little use in communicating the quantitative aspects of the data, but they can highlight qualitative results.

Ehrenberg's advice was conceived within the traditional paradigm of an author preparing a printed report. How do they apply to Computer-based Message Systems?

In the case of a staff manager using a CBMS as a consistent interface to access specialised packages, there is a clear need for some simple, powerful editing functions, over and above those available in standard text editing, which will enable him to apply Ehrenberg's recommendations (or some similar principles) to the output. Indeed, the facility to try out different presentations of the data may not only be needed as a support to his report-writing, but as an essential part of his creative "information massage". So to his online editor a table should be not merely a piece of text but a structured data object, to which he can apply such functions as:

round
unround
collapse
transpose columns and rows
re-order columns/rows
group columns/rows on criterion

Some limited calculation functions should also be provided, eg for insertion of marginals not output by the specialist package.

Not only should the table be structured data to the editor; when the staff manager inserts it into a report and mails it to line or top management, the table should still retain its structure. When the message is initially displayed to the recipient, he should see tables in the format that the author has decided; thus the data can be presented in the format the staff manager judges to be most appropriate for his communicative purpose. But the line-manager can still retain the option to extract a greater degree of detail from the aggregates presented to him, without necessarily having to work with the original database used by the staff manager. The line manager should also have access to the facilities for information massage provided for the staff function, though it is not to be expected that he will use them to the same extent.

In the design of a management-oriented system, some care must be taken to solve the problem of reconciling interactiveness and flexibility with achievement of good format standards. One solution would be to develop a "house style" of table-formats which are supported by readily-available functions (eg by menu window or on soft function keys), while non-standard formats could be achieved only by a much more complex editing procedure. Clearly there must be a great deal of user-involvement in developing the house-style, otherwise there is a risk that the restrictions will be experienced as unacceptable.

Supporting the Decision Process and Implementation

Decision-support Systems (Carlson, 1977; Keen & Wagner, 1979; Kroeber, Watson & Sprague, 1980) have generally been conceived principally as giving executives rapid access to, and control over, formal models of the organizational environment. However, "the ever increasing volume and variety of textual and other non-numeric information needed for management decision-making" has been identified by Fereira & Collins (1979) as a major challenge to the Management Information Systems function. Organizations selling in high-technology markets, for example, must take account not only of the state of the art in technology itself, but also of social and political factors of three distinct kinds: regulatory constraints, dominant supplier policies, and currency turbulence (Newman & Ward, op.cit.). This phase requires facilities for creation, manipulation and retrieval of text and graphics; easy incorporation of numeric, graphic and text data into a single report; facilities to support indexing, cataloguing, mailing, updating and support for collaborative report generation by authors at the same or remote sites, either synchronous or asynchronous.

Obtaining Feedback on Decision-Consequences

Computer conferencing systems are particularly suitable for obtaining feedback on decision-consequences, providing asynchronous communication, which frees the communicators from the time-zone constraints inherent in face to face or telephone discussions. Extensive experiments on computer conferencing by Vallee, Turoff, and others have established that it is possible to facilitate effective group communication amongst people who are scattered over wide distances (cf. Hiltz & Turoff, 1978; Vallee et al 1974, 1975; Vallee 1976; Vallee & Gibbs, 1976; Zinn, 1977). Uhlig (1977) has described CBMS as "The Great Equaliser", leading to great candour. Use of written rather than verbal communication for feedback means that a permanent record can be readily maintained, if required.

DISTRIBUTED INTELLIGENCE REQUIREMENTS

Existing Computer-based Message Systems tend either to be too decentralised or too centralised (Sandberg, 1979). The human-factors considerations put forward above suggest that a rather high degree of intelligence in the terminals will be desirable, if Computer-based Message systems are to be acceptable to management. (Compare for example the use of intelligent terminals in the SIGMA system - Stotz et al 1979; also Henne, 1980). However, much of the Decision-Support capability of the systems here envisaged will depend upon access to centralised databases. Managers will need to be shielded from the complexities of knowing where their data is resident at any particular time during the interaction, without loss of flexibility or power.

THE USER SUPPORT FUNCTION

The user support function for management-oriented CBMS will not only be crucial in its initial acceptance, but will be likely to grow and develop, rather than wither away, as managers gain experience and confidence in the system (cf. Damodaran, 1976). While the system should incorporate an online tutor (cf. Grignetti, 1975; Rothenburg, 1979), as well as full online documentation and a Help facility, access to expert human advice on the system's capabilities will be an ongoing requirement, which should be planned for in the early stages. Such advice could take the form of designated "local experts" as well as systems staff.

CONCLUSIONS

Computer-based Message Systems have considerable potential as a management tool. However, in order to satisfy Human Factors requirements CBMS will have to attain very ambitious targets. Pure electronic mail or electronic meetings, while easier to implement, will have limited application to the most pressing needs of management.

REFERENCES

(1) Argenti, J (1976) Corporate Collapse. Maidenhead: McGraw-Hill.
(2) Ashby, W R (1956) Introduction to Cybernetics. London: Chapman & Hall.
(3) Bell, D (1973) The Coming of Post-Industrial Society. New York: Basic Books.
(4) Carlson, E D (ed 1977) "Proceedings of a Conference on Decision Systems": Database, Vol.8, No.3
(5) Carlson, E D and Morgan, H L (1977) "Conference on Decision Support Systems"; Comm.A.C.M., 20, 361.
(6) Copeman, G, Luijk, H and de Hanika, F (1963) How the executive spends his time, London: Business Publications
(7) Drucker, P (1969) The Age of Discontinuity. London: Heinemann.
(8) Eason, K D (1976a) "Understanding the naive computer user"; Computer Journal, 19, 1, 3-7
(9) Eason, K D (1976b) "A Task-Tool analysis of manager-computer interaction, Mati, Greece, 1976
(10) Ehrenberg, A S C (1977) "Rudiments of Nueracy"; Journal of the Royal Statistical Society, Series A, 1940(3), 277-297, 307-323.
(11) Emery, F E (1977) Futures We Are In. Leiden: Martinus Nijhoff.

(12) Emery, F E & Trist, E L (1965) "The Casual Texture of Organizational Environments"; Human Relations, 18, 21-31.
(13) Engelbart, D C (1975) "NLS Teleconferencing features" SRI-ARC 33076, Menlo Park, CA.
(14) Engelbart, D C (1979) "Consistency in NLS"; Workshop on Man-Computer Communications, EIFIP, Le Bischenberg, Strasbourg, France
(15) Engelbart, D C, Watson, R W & Norton, J C (1973) "The Augmented Knowledge Workshop" SRI-ARC 14724, Menlo Park, CA.
(16) Fereira, J & Collins, J F (1979) "The Changing Role of the MIS Executive"; Datamation, 25(13), 26-32.
(17) Galbraith, J K (1974) The New Industrial State. 2nd ed, Harmondsworth: Penguin.
(18) Grignetti, J (1975) "NLS-SCOLAR" AFIPS Conference Proceedings Vol 44 1975 National Computer Conference
(19) Henne, P (1980) "KOMEX: An experimental computer-conference system"; in Integrated Office Systems: Burotique ed N Naffah. Amsterdam: North-Holland
(20) Hiltz, S R & Turoff, M (1978) The Network Nation. New York: Addison-Wesley.
(21) Horne, J H & Lupton, T (1965) "The Work Activities of Middle Managers" Journal of Management Studies, 1(2), 14-33.
(22) Hough, R W & Panko, R R (1977) "Teleconfercing Systems: A State of the Art Survey and Preliminary Analysis". Menlo Park, CA: SRI.
(23) Keen, P G & Wagner, G R (1979) "DSS: An Executive Mind-Support System"; Datamation, 25, 12, 117-122.
(24) Kroeber, D W, Watson, H J & Sprague, R H (1980) "An Empirical Investigation and Analysis of the Current State of Information Systems Evolution"; Information and Management 3, 35-43.
(25) Machlup, F. (1962) The Production and Distribution of Knowledge in the USA Princeton: Princeton University Press.
(26) Macintosh, N B & Daft, R L (1978) "User Department Technology and Information Systems Design"; Information and Management 1, 123-131.
(27) Mintzberg, H (1973) The Nature of Managerial Work New York, NY: Harper & Row.
(28) Newman, J C (1976) "Human Factors Input Vital to Systems Thinking"; Computer Weekly, 520, 22-23.
(29) Newman, J C (1978) "Faster Extraction of Numerical Information from One-dimensional Computer Printout Tables". British Psychological Society Annual Conference, University of York, U.K.
(30) Newman, J C & Ward, T B (1980) "The Organizational Context of Office Systems"; in Integrated Office Systems: Burotique ed N Naffah. Amsterdam: North-Holland.
(31) Perrow, C (1967) "A Framework for the Comparative Analysis of Organizations", American Sociological Review, 194-208.
(32) Pye R & Williams, E (1977) "Teleconferencing: Is Video Valuable or is Audio Adequate?"; Telecommunications Policy, 1, 230-241.
(33) Rohlfs, S (1979) "User Interface Requirements" in Convergence: Computers, Communications and Office Automation, Vol.2, Maidenhead, UK: Infotech.
(34) Rothenberg, J (1979) "Online Tutorials and Documentation for the SIGMA message service" AFIPS Conference Proceedings Vol 48 1979 National Computer Conference 863-868.
(35) Sandberg, A (1979) "Electronic Mail and Message Systems" in Convergence: Computers, Communications and Office Automation Vol.2, Maidenhead, UK: Infotech.
(36) Stewart, R (1970) Managers and their Jobs. London: Pan.

(37) Stewart, R (1976) Contrasts in Management. Maidenhead: McGraw-Hill, UK.
(38) Stewart, T F M (1976a) "Displays and the Software Interface". Applied Ergonomics, 7(3), 137-146.
(39) Stewart, T F M (1976b) "The Specialist User" NATO Advanced Study Institute on Man-Computer Interaction, Mati, Greece, 1976.
(40) Stotz, R, Raveling, P & Rothenburg, J (1979) "The terminal for the Military Message Experiment" AFIPS Conference Proceedings Vol 48 1979 National Computer Conference.
(41) Stotz, R, Tugender, R, Wilczynski, D & Oestriecher, D (1979) "SIGMA - An Interactive Message Service for the Military Message Experiment" AFIPS Conference Proceedings Vol 48 1979 National Computer Conference, 839-846.
(42) Terreberry, S (1968) "The Evolution of Organizational Environments"; Administrative Science Quarterly, 12, 590-613.
(43) Uhlig, R (1977) "Human Factors in Computer Message Systems"; Datamation 23(5), 120-126.
(44) Vallee, J (1976) "The FORUM Project: Network Conferencing and its Future Applications"; Computer Networks, 1, 39-52.
(45) Vallee, J & Gibbs B (1976) "Distributed Management of Scientific Projects"; Telecommunications Policy, 1, 75-85.
(46) Vallee, J and IFTF Staff (1974, 1975) Group Communication through Computers (3 vols) Menlo Park, CA: Institute for the Future.
(47) Ward, T B & Newman, J C (1980) "Computer Conferencing, Regional Policy and Post-Industrial Society". RMCA Conference on Regional Policy, Welsh Regional Management Centre, May 1980.
(48) Wright, P (1977) "Decision-making as a factor in the ease of using numerical tables". Ergonomics, 20, 91-96.
(49) Zinn, K (1977) "Computer facilitation of communication within professional communities"; Behavioral Research Methods and Instrumentation, 9, 96-107.

ISBN 0 444 85253 6